

Modellbildung und Fehlerdiagnose mit rekurrenten Fuzzy-Systemen

Vom Fachbereich
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades
eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

Dipl.-Ing. Andreas Schwung

geboren am 17. Januar 1982 in Hamm

Referent:	Prof. Dr.-Ing. J. Adamy
Korreferent:	Prof. Dr.-Ing. A. Kroll
Tag der Einreichung:	6. Dezember 2010
Tag der mündlichen Prüfung:	11. Februar 2011

D17
Darmstadt 2010

Vorwort

Die vorliegende Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Automatisierungstechnik, Fachgebiet Regelungstheorie und Robotik, der TU Darmstadt.

Mein besonderer Dank geht an Herrn Prof. Dr.-Ing. Jürgen Adamy, der mir die Möglichkeit gegeben hat, diese Arbeit unter seiner Betreuung anfertigen zu können. Seine Unterstützung, aber auch der nötige Freiraum sowie das hervorragende Arbeitsklima haben sehr zum Gelingen der Arbeit beigetragen.

Weiterhin möchte ich mich bei Prof. Dr.-Ing. Andreas Kroll für die freundliche Übernahme des Korreferats bedanken.

Bedanken möchte ich mich auch bei allen Kollegen am Fachgebiet, insbesondere dem Kollegen von „schräg gegenüber“ Thomas Gußner, für die herausragende Zusammenarbeit und die vielen Diskussionen rund um die Forschung und darüber hinaus. Das tolle Arbeitsklima am Fachgebiet und das freundschaftliche Verhältnis untereinander ist einmalig.

Für die kritische Durchsicht der Arbeit danke ich Thomas Gußner, Klaus Kefferpütz und Arne Wahrburg ganz herzlich. Valentina Ansel danke ich für die Erstellung der 3D-Bilder. Großer Dank geht auch an alle Studenten, insbesondere an Andreas Ortseifen, die mit ihren Beiträgen zum Gelingen der Arbeit beigetragen haben.

Nicht zuletzt möchte ich meinen Eltern und meinem Bruder für die stetige und vielseitige Unterstützung in den vergangenen Jahren herzlich danken.

Inhaltsverzeichnis

Abkürzungen und wichtige Symbole	VIII
Kurzfassung/Abstract	XII
1 Einführung	1
1.1 Ziele der Arbeit	3
1.2 Aufbau der Arbeit	4
2 Modellbildung mit Fuzzy-Systemen: ein Überblick	6
2.1 Fuzzy-Modellstrukturen	6
2.2 Modellbildung mit Fuzzy-Systemen	15
2.2.1 Wissensbasierte und datengestützte Modellbildung .	16
2.2.2 Strukturoptimierung	16
2.2.3 Parameteroptimierung	21
2.2.4 Erhalt der Interpretierbarkeit	22
3 Grundlagen	25
3.1 Zeitkontinuierliche rekurrente Fuzzy-Systeme	25
3.2 Zeitdiskrete rekurrente Fuzzy-Systeme	30
3.3 Darstellung rekurrenter Fuzzy-Systeme als hybride Systeme	33
3.3.1 Zerlegung des RFS in elementare Hyperquader . . .	34
3.3.2 Darstellung eines elementaren Hyperquaders	36
3.3.3 Darstellung des gesamten RFS als hybrides System .	39
3.3.4 Diskussion der Darstellung	39
4 Stabilitätsanalyse rekurrenter Fuzzy-Systeme	41
4.1 Grundlagen	42
4.1.1 Stabilität hybrider Systeme	42
4.1.2 SOS-Methodik	43
4.2 Stabilität bekannter Ruhelagen	46
4.3 Stabilität unbekannter Ruhelagen	47
4.4 Beispiele	56
4.4.1 Zeitkontinuierliches System	56

4.4.2	Populationsdynamik	57
4.5	Zusammenfassung und Fazit	58
5	Modellierung mit zeitkontinuierlichen rekurrenten Fuzzy-Systemen	60
5.1	Einführung und Problemdefinition	60
5.2	Parameteroptimierung	63
5.2.1	Optimierung der Kernpositionsableitungen	63
5.2.2	Optimierung der Kernpositionen	74
5.2.3	Gleichzeitige Optimierung aller Parameter	77
5.3	Strukturoptimierung	79
5.3.1	Partitionierung des Eingangsraums	80
5.3.2	Reduktion der Kernpositionsableitungen	84
5.3.3	Reduktion der Regeln	85
5.4	Auswahl der relevanten Eingangs- und Zustandsgrößen . . .	87
5.5	Simulationsbeispiele	90
5.5.1	Bioreaktor	90
5.5.2	Rührkesselreaktor	96
5.6	Zusammenfassung und Fazit	101
6	Modellierung mit zeitdiskreten rekurrenten Fuzzy-Systemen	103
6.1	Einführung und Problemdefinition	103
6.2	Identifikation der Regelbasis mit ganzzahliger Optimierung	107
6.2.1	Ganzzahlige Optimierung mit Nebenbedingungen . .	110
6.3	Parameteroptimierung	112
6.4	Beispiel: Thermofluidischer Prozess	114
6.5	Zusammenfassung und Fazit	119
7	Fehlerdiagnose in dynamischen Systemen	120
7.1	Einführung	120
7.2	Residuengenerierung	123
7.3	Fehlerisolation	126
7.3.1	Fehlerisolation mit statischen Fuzzy-Systemen	127
7.3.2	Fehlerisolation mit zeitdiskreten rekurrenten Fuzzy-Systemen	128
7.4	Beispiel: Dreitank-System	137
7.5	Zusammenfassung und Fazit	143
8	Anwendungen	144
8.1	Modellierung einer Destillationskolonne	145

8.1.1	Systembeschreibung	146
8.1.2	Modellierung der Kolonne mit RFS	148
8.1.3	Fazit	157
8.2	Fehlerdiagnose an einer hydraulischen Servoachse	160
8.2.1	Systembeschreibung	161
8.2.2	Beschreibung der Fehlerfälle	163
8.2.3	Fehlerdiagnose	167
8.2.4	Fazit	176
9	Zusammenfassung	177
A	Berechnungen	180
A.1	Singulärwertzerlegung zur Lösung von Least-Squares Problemen	180
A.2	Matrizendarstellung eines Hyperquaders	181
B	Modellparameter	183
B.1	Parameter und Einstellwerte der Fuzzy-Modelle	183
B.2	Symptomtabelle für die Doppelfehler	186
B.3	Datensätze der Simulationsmodelle	187
	Literaturverzeichnis	193

Abkürzungen und wichtige Symbole

Abkürzungen

APRBS	amplitudenmoduliertes Pseudo-Rausch-Binär-Signal
ASK	Ausgangsstrukturkriterium
DRFS	zeitdiskretes rekurrentes Fuzzy-System
EA	Ein-/Ausgangsdarstellung
ESK	Eingangsstrukturkriterium
FDI	Fehlerdetektion und Isolation
KRFS	zeitkontinuierliches rekurrentes Fuzzy-System
LS	Least-Squares
MFS	Mamdani Fuzzy-System
MOP	Mehrzieloptimierungsproblem
MSE	Mean Square Error (mittlerer quadratischer Fehler)
NFS	Neuro-Fuzzy-System
NGGOP	nichtlineares gemischt-ganzzahliges Optimierungsproblem
OLS	orthogonale Least-Squares
QP	quadratische Programmierung
RFS	rekurrentes Fuzzy-System
RMSE	Root Mean Square Error (Wurzel aus MSE)
SFS	statisches Fuzzy-System
SOS	Sum-of-Squares
TSK	Takagi-Sugeno-Kang
ZR	Zustandsraum

Symbole

Es werden zuerst die lateinischen und dann die griechischen Symbole jeweils alphabetisch aufgelistet. Dabei sind nur solche Symbole aufgeführt, die über

die gesamte Arbeit hinweg und nicht nur in einem Abschnitt verwendet werden.

$a_{ij}(\mathbf{x}), b_{ij}(\mathbf{x})$	Hilfspolynome
$c_{ij}(\mathbf{x}), d_j(\mathbf{x})$	Hilfspolynome
\mathbf{a}_{w_i}	Selektionsvektor
\mathbf{b}_{nb}	Vektor für Nebenbedingungen bei binärer Optimierung
$\mathbf{f}(\mathbf{x}, \mathbf{u})$	Dynamikfunktion des RFS
\mathbf{f}_l	polynomiales Vektorfeld des l -ten Hyperquaders
$\mathbf{g}(\mathbf{x}, \mathbf{u})$	Ausgangsfunktion des RFS
i	Index der Zustände
\mathbf{j}	Indexvektor der linguistischen Zustände
m	Anzahl der Eingangsgrößen
n	Anzahl der Zustandsgrößen
p	Index der Eingänge
\mathbf{q}	Indexvektor der linguistischen Eingänge
\mathbf{r}	Residuenvektor
$\mathbf{r}_l(\mathbf{x}, \mathbf{u})$	Grenzen eines Hyperquaders
$r_x(i)$	Zahl der Zugehörigkeitsfunktionen des Zustands x_i
$r_u(p)$	Zahl der Zugehörigkeitsfunktionen des Zustands u_p
$s_{j_i}^{x_i}$	Kernposition des Zustands x_i mit Index j_i
$s_{q_p}^{u_p}$	Kernposition des Eingangs u_p mit Index q_p
$s_{w_i}^{\dot{x}_i}$	Kernpositionsableitung der Ableitung \dot{x}_i mit Index w_i
$\mathbf{s}_j^{\mathbf{x}}, \mathbf{s}_q^{\mathbf{u}}, \mathbf{s}_{\mathbf{v}(\mathbf{j}, \mathbf{q})}^{\mathbf{y}}$	Kernpositionsvektoren der Zustände, Eingänge und Ausgänge
$\mathbf{s}_{\mathbf{w}(\mathbf{j}, \mathbf{q})}^{\dot{\mathbf{x}}}$	Vektor der Kernpositionsableitungen
$\mathbf{s}^{\dot{x}_i}$	Vektor aller Kernpositionsableitungen in x_i -Richtung
u_q	q -te Eingangsvariable
(v)	Laufindex der Messdaten
\mathbf{w}	Indexvektor der linguistischen Zustandsableitungen (KRFS) bzw. Ausgangszustände (DRFS)
\mathbf{x}_e	Ruhelage
x_i	i -te Zustandsvariable
\dot{x}_i	i -te Zustandsableitung
\mathbf{z}_i	Ausgangsvektor bei LS-Optimierung
A	Matrix der Ungleichungsnebenbedingungen
A_{itp}	Matrix der Interpolationsnebenbedingungen
A_{nb}	Matrix der Nebenbedingungen bei binärer Optimierung
\mathbf{B}^{x_i}	Matrix zur Definition der Gleichungsnebenbedingungen
\mathbf{G}	Gradient der Gütefunktion

J_i	Gütefunktion der i -ten Ableitung
K	Anzahl der Hyperquader
L	linguistischer Wert
L^{x_i}	linguistischer Wert des Zustands x_i mit Index j_i
$L_{q_p}^{j_i}$	linguistischer Wert des Eingangs u_p mit Index q_p
$L_{w_i}^{x_i}$	linguistischer Wert der Ableitung \dot{x}_i mit Index w_i
$\mathbf{L}_j^{\mathbf{x}}, \mathbf{L}_q^{\mathbf{u}}, \mathbf{L}_w^{\dot{\mathbf{x}}}$	linguistische Vektoren der Zustände \mathbf{x} , Eingänge \mathbf{u} und Zustandsableitungen $\dot{\mathbf{x}}$
N	Anzahl der Messdaten
R	Anzahl der Regeln
R_w	gewünschte Zahl an Regeln
$V(\mathbf{x})$	Lyapunov-Funktion
\mathcal{X}	Zustandsraum
\mathcal{Z}	Eingangs-/Zustandsraum
\mathcal{Z}_{in}	Interpolationsregion des RFS
\mathcal{Z}_{ex}	Extrapolationsregion des RFS
\mathcal{Z}_l	Eingangs-/Zustandsraum des l -ten Hyperquaders
$\alpha_{\text{ASK}}, \alpha_{\text{ASK}}^*$	Parameter ASK
α_{EA}	Grenzwert bei Rückwärtsauswahl
α_{gr}	Grenzwert für Optimierung mit Gradientenverfahren
β^*	Parameter ESK
γ	Zahl der Cluster im ASK
γ_0	Parameter ASK
δ_w	Distanzmaß
$\delta_0, \delta_1, \delta^*$	Parameter bei Regelreduktion
$\varepsilon(\mathbf{x})$	positiv definites Hilfspolynom
η	Schrittweite
κ	Index der Iterationsschritte
$\mu_{j_i}^{x_i}$	Zugehörigkeitsfunktion des linguistischen Wertes $L_{j_i}^{x_i}$
ξ	Optimierungsparameter des Gradientenverfahren
ρ_{ESK}	Parameter im ESK
ρ_i	Symptom
σ	Varianz der Ausgangsdaten
τ_i	Gewichtungsfaktoren für MOP
φ_i	Fehler i
$\boldsymbol{\varphi}$	Fehlervektor
\mathbf{v}_i	Vektor der binären Optimierungsparameter
$\boldsymbol{\omega}$	Monomvektor
Γ	Matrixdarstellung der Prämissen eines Hyperquaders

Θ_l	Koeffizientenmatrix des l -ten Hyperquaders
$\Lambda_i(s_{w^*}^{\dot{x}_i})$	Menge der zur Kernpositionsableitung $s_{w^*}^{\dot{x}_i}$ gehörenden Regeln
$\Xi_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}, \mathbf{u})$	Prämisse einer Regel
$\Xi_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}, \mathbf{u})$	Vektor der Prämissen aller Regeln
$\Sigma[\mathbf{x}]$	Menge der SOS-Polynome
Υ	Regressionsmatrix bei binärer Optimierung
Ψ	Regressionsmatrix bei LS-Optimierung

Kurzfassung/Abstract

Die vorliegende Arbeit beschäftigt sich mit der Stabilitätsanalyse, Modellierung und Fehlerdiagnose mit rekurrenten Fuzzy-Systemen. Hierbei werden sowohl zeitkontinuierliche als auch zeitdiskrete rekurrente Fuzzy-Systeme betrachtet.

Zunächst wird ausgehend von der Darstellung rekurrenter Fuzzy-Systeme als hybride polynomiale Systeme ein Verfahren zur Stabilitätsanalyse vorgestellt. Die polynomiale Darstellung ermöglicht die Anwendung von Verfahren der konvexen Optimierung, sodass die Stabilität effizient untersucht werden kann. Anschließend wird die Anwendung rekurrenter Fuzzy-Systeme zur Modellierung dynamischer Systeme betrachtet. Es werden Methoden vorgestellt, mit denen auf Basis von Datensätzen sowie verschiedener Formen von Expertenwissen rekurrente Fuzzy-Systeme automatisiert entworfen werden können. Das Ziel ist es, neben der guten Nachbildung des Systemverhaltens insbesondere qualitative und linguistisch interpretierbare Modelle zu erzeugen. Eine Anwendung der entwickelten Verfahren zur Modellbildung stellt die Fehlerdiagnose in technischen Systemen dar. Zeitkontinuierliche rekurrente Fuzzy-Systeme eignen sich hier insbesondere zur modellbasierten Fehlerdetektion, während zeitdiskrete rekurrente Fuzzy-Systeme vor allem zur Fehlerisolation eingesetzt werden.

Anwendungsbeispiele aus der chemischen Industrie und dem Bereich der hydraulischen Servoantriebe zeigen die Umsetzung der vorgestellten Methoden in die Praxis.

Modeling and Fault Diagnosis using Recurrent Fuzzy Systems

This work deals with stability analysis, modeling and fault diagnosis with recurrent fuzzy systems. Both continuous-time and discrete-time recurrent fuzzy systems are considered.

Based on the representation of recurrent fuzzy systems as hybrid polynomial systems an approach for stability analysis is introduced. The polynomial representation allows an efficient stability analysis by means of convex optimization. Afterwards, modeling of dynamical systems using recurrent

fuzzy systems is discussed. Using measurement data and different forms of expert knowledge methods allowing the automated design of recurrent fuzzy systems are presented. Beside a good approximation of the dynamic behavior, the main aim is the design of qualitative and linguistically interpretable models. An application of the presented methods for modeling is fault diagnosis of technical systems. Here, continuous-time recurrent fuzzy systems are used for residual generation, while discrete-time recurrent fuzzy systems are especially suitable for fault isolation.

Application examples arising from the chemical industry and hydraulic servo systems underline the practical applicability of the presented methods.

1 Einführung

Modelle als Abstraktion der realen Welt werden in verschiedensten wissenschaftlichen Disziplinen wie der Wirtschaftstheorie, den Naturwissenschaften und den Ingenieurwissenschaften verwendet, um bestimmte Phänomene oder Systemzusammenhänge mathematisch beschreiben zu können. Eingesetzt werden sie hierbei unter anderem zur Systemanalyse, Systemoptimierung, Prognose, Prozessüberwachung und -regelung sowie zur Fehlerdiagnose. Der Generierung entsprechender Modelle, auch als Modellbildung bezeichnet, kommt deshalb entscheidende Bedeutung zu.

Grundsätzlich kann man zur Modellbildung zwei Wege beschreiten [78]. Der eine besteht in der theoretischen Analyse des Systems, beispielsweise in Form von Bilanzgleichungen oder phänomenologischen Gleichungen. Im Ergebnis erhält man dann ein „White-Box“-Modell. Typische Beispiele für solche „White-Box“-Modelle sind mechanische Systeme, bei denen die Zusammenhänge durch gewöhnliche, bei örtlich verteilten Systemen auch durch partielle Differentialgleichungen hinreichend genau beschrieben werden können. Die zweite Möglichkeit besteht in der experimentellen Analyse des Systems, die auf Messdaten des zu analysierenden Prozesses basiert. Dies führt auf ein „Black-Box“-Modell, bei dem im Gegensatz zum „White-Box“-Modell nur noch das Ein-/Ausgangsverhalten des Prozesses, nicht aber die internen Zusammenhänge der Prozessgrößen, modelliert wird.

In der Realität finden sich allerdings eine Reihe von Systemen, die weder vollständig bekannt noch vollständig unbekannt sind. Solche Systeme stellen eine Mischform dar und werden deshalb als „Grey-Box“-Modelle bezeichnet [69, 109]. Abhängig davon, wieviel über den Prozess bekannt ist, lassen sich verschiedene Abstufungen unterscheiden. Modelle, bei denen nur einzelne Parameter in den Differentialgleichungen unbekannt sind, werden als „Light-Grey“-Modelle bezeichnet. Sind dagegen Teile des Prozesses auch in ihrer Struktur nicht bekannt, spricht man von „Dark-Grey“-Modellen. Eine weitere Form von „Grey-Box“-Modellen stellen Systeme dar, bei denen neben Messdaten auch linguistisches Wissen in Form von Regeln von erfahrenen Prozessbedienern vorhanden ist. Typische Beispiele sind chemische und biotechnologische Prozesse. Für solche Anwendungen sind

Fuzzy-Systeme besonders geeignet [154, 155], da sie sowohl die formulierten Regeln explizit berücksichtigen können als auch eine datengestützte Modellbildung erlauben.

Fuzzy-Systeme in ihrer ursprünglichen Form können nur statische Zusammenhänge nachbilden. Um auch dynamische Systeme modellieren zu können, müssen Rückführungen der Ausgangs- oder Zustandsgrößen hinzugefügt werden. Eine Klasse von Fuzzy-Systemen, die eine solche Struktur aufweisen, sind rekurrente Fuzzy-Systeme (RFS), die sowohl im Zeitdiskreten [5, 98] als auch im Zeitkontinuierlichen [4, 49] definiert sind. Eine erste Anwendung zeitdiskreter rekurrenter Fuzzy-Systeme (DRFS) ist die Durchbrucherkennung in Stranggießanlagen [2]. Ein erstes Beispiel zur Modellbildung wurde in [50, 97] mit der Simulation von Verkehrssituationen vorgestellt. Auch zeitkontinuierliche rekurrente Fuzzy-Systeme (KRFS) wurden bereits zur Modellbildung verwendet. So wurde in [49] das dynamische Verhalten einer Hochtemperatur-Brennstoffzelle durch ein KRFS modelliert. Allerdings ist allen bisherigen Ansätzen zur Modellierung dynamischer Systeme mit RFS gemein, dass die Herleitung der Modelle im Gegensatz zu verschiedenen anderen Ausprägungen von dynamischen Fuzzy-Systemen (vgl. Kapitel 2) allein auf Expertenwissen beruht. Dies führt zu einem oftmals langwierigen Auslegungsprozess mit vielen Iterationen, insbesondere wenn das Modell mit Parametern quantifiziert werden soll. Des Weiteren existieren eine Reihe von Systemen, bei denen das Expertenwissen für eine vollständige Modellbildung nicht ausreicht. Somit ist es wünschenswert, zusätzlich zum Expertenwissen auch Messdatensätze des Systems zur Modellbildung zu nutzen und damit den Entwurfsprozess zu automatisieren.

Ein weiteres bekanntes Anwendungsgebiet von Fuzzy-Systemen ist die Fehlerdiagnose [54, 80], die allgemein in die Teilschritte Fehlerdetektion und Fehlerisolation gegliedert werden kann. Hierbei dient die Fehlerdetektion der Erkennung von Abweichungen des Prozesses vom normalen Betriebszustand. Da zunehmend modellbasierte Verfahren eingesetzt werden, ist die Fehlerdetektion eng mit der Modellierung dynamischer Systeme verbunden. Hat man nämlich den Prozess nachgebildet, beispielsweise durch ein RFS, so kann man von Abweichungen zwischen den Messgrößen des Prozesses und dem Modell auf einen aufgetretenen Fehler schließen. Die anschließende Fehlerisolation bestimmt dann, welcher Fehler aufgetreten ist. Bisher werden hierzu zumeist statische Fuzzy-Systeme (SFS) eingesetzt. Mit diesen ist es allerdings im Allgemeinen nicht möglich, dynamische Fehlerverläufe sowie mehrere auftretende Fehler eindeutig zu isolieren. Hier

haben wiederum RFS Vorteile, da sie auch dynamisches Verhalten nachbilden können, sodass obige Fehlerfälle abgedeckt werden.

1.1 Ziele der Arbeit

Wie bereits angedeutet wurde die Modellbildung mit rekurrenten Fuzzy-Systemen bisher nur an Beispielen gezeigt. Eine systematische Untersuchung, wie eine automatisierte Modellbildung unter Berücksichtigung verschiedener Formen von Expertenwissen und Messdaten erfolgen kann, existiert bisher nicht. Darüber hinaus ist die Anwendung von RFS zur Fehlerdiagnose bisher noch nicht untersucht worden. Diese Lücke soll im Rahmen dieser Arbeit geschlossen werden. Als dritter Punkt wird aufbauend auf der Beschreibung RFS als hybride Systeme ein Ansatz zur Stabilitätsanalyse entwickelt, der die bisher bekannten Verfahren ergänzt und erweitert. Zusammenfassend lassen sich somit die folgenden Ziele ableiten:

- Entwicklung von Verfahren zur Modellierung zeitkontinuierlicher und zeitdiskreter dynamischer Systeme mithilfe von RFS: Es sollen Verfahren entwickelt werden, die sowohl die Optimierung der Systemstruktur als auch der Systemparameter ermöglichen. Die Optimierung soll datengestützt erfolgen, wobei das mit unterschiedlichem Detaillierungsgrad verfügbare, qualitative Vorwissen möglichst umfassend im Modellierungsprozess berücksichtigt werden soll. Im Ergebnis sollen sich interpretierbare, regelbasierte Modelle dynamischer Systeme ergeben, die über das reine Ein-/Ausgangsverhalten hinaus auch interne Zusammenhänge zwischen Prozessgrößen abbilden können.
- Entwicklung von Verfahren zur Fehlerdiagnose mit RFS: Hierbei sollen die beiden Kernschritte der Fehlerdiagnose, namentlich die Fehlerdetektion und die Fehlerisolation, mithilfe von RFS durchgeführt werden. Zur Fehlerdetektion soll ein auf RFS basierender Ansatz zur modellbasierten Residuengenerierung entwickelt werden, der die entwickelte Modellbildungsmethodik verwendet. Zur Fehlerisolation sollen die bekannten Ansätze mit statischen Fuzzy-Systemen auf zeitdiskrete RFS erweitert werden, was insbesondere die Isolation dynamischer Fehler und mehrerer aufeinander folgender Fehler ermöglichen soll.
- Einführung einer alternativen Beschreibung rekurrenter Fuzzy-Systeme als hybride polynomiale Systeme: Auf Basis der neuen Sys-

tembeschreibung soll ein neuer Ansatz zur Stabilitätsanalyse von rekurrenten Fuzzy-Systemen entwickelt werden, der die bisherige Analysemethodik erweitert und die Analyse von zeitdiskreten und zeitkontinuierlichen RFS in einem gemeinsamen Rahmen ermöglicht.

- Die Anwendung der Verfahren und deren Praxistauglichkeit zur Modellbildung und Fehlerdiagnose soll anhand zweier Beispiele illustriert werden.

1.2 Aufbau der Arbeit

Abbildung 1.1 zeigt eine Übersicht über den Aufbau der Arbeit und die Zusammenhänge zwischen den Themengebieten. Nach der Einführung gibt Kapitel 2 einen Überblick über die Modellierung dynamischer Systeme mit Fuzzy-Systemen, um die vorliegende Arbeit in den Stand der Technik einzuordnen. Daran anschließend werden in Kapitel 3 die Grundlagen rekurrenter Fuzzy-Systeme vorgestellt. Hierbei wird sowohl auf zeitdiskrete als auch auf zeitkontinuierliche rekurrente Fuzzy-Systeme eingegangen. Darüber hinaus wird eine neue, alternative Systembeschreibung eingeführt, die rekurrente Fuzzy-Systeme als hybride polynomiale Systeme darstellt.

Auf dieser Darstellung aufbauend werden in Kapitel 4 die Stabilitätseigenschaften rekurrenter Fuzzy-Systeme untersucht. Es wird ein neuer Ansatz vorgeschlagen, der die polynomiale Systembeschreibung verwendet und so die Untersuchung zeitdiskreter und zeitkontinuierlicher rekurrenter Fuzzy-Systeme in einem gemeinsamen Rahmen ermöglicht.

Kapitel 5 befasst sich dann mit der Modellierung dynamischer Systeme mithilfe zeitkontinuierlicher rekurrenter Fuzzy-Systeme. Insbesondere wird gezeigt, wie auf Basis von Prozesswissen und aus dem System gewonnener Datensätze technische Prozesse durch interpretierbare Modelle nachgebildet werden können.

In Kapitel 6 wird die Modellbildung mit zeitdiskreten rekurrenten Fuzzy-Systemen betrachtet. Aufgrund des speziellen Aufbaus eignen sich zeitdiskrete rekurrente Fuzzy-Systeme vor allem zur Gewinnung qualitativer Systemmodelle sowie zur Mustererkennung. Auch hier werden Verfahren vorgestellt, mit denen Modelle automatisiert gebildet werden können.

Eine erste Anwendung der in den vorigen Kapiteln vorgestellten Verfahren zur Modellbildung erfolgt in Kapitel 7 am Beispiel der Fehlerdiagnose in technischen Systemen. Zeitkontinuierliche rekurrente Fuzzy-Systeme eignen sich hier insbesondere zur modellbasierten Fehlerdetektion, wäh-

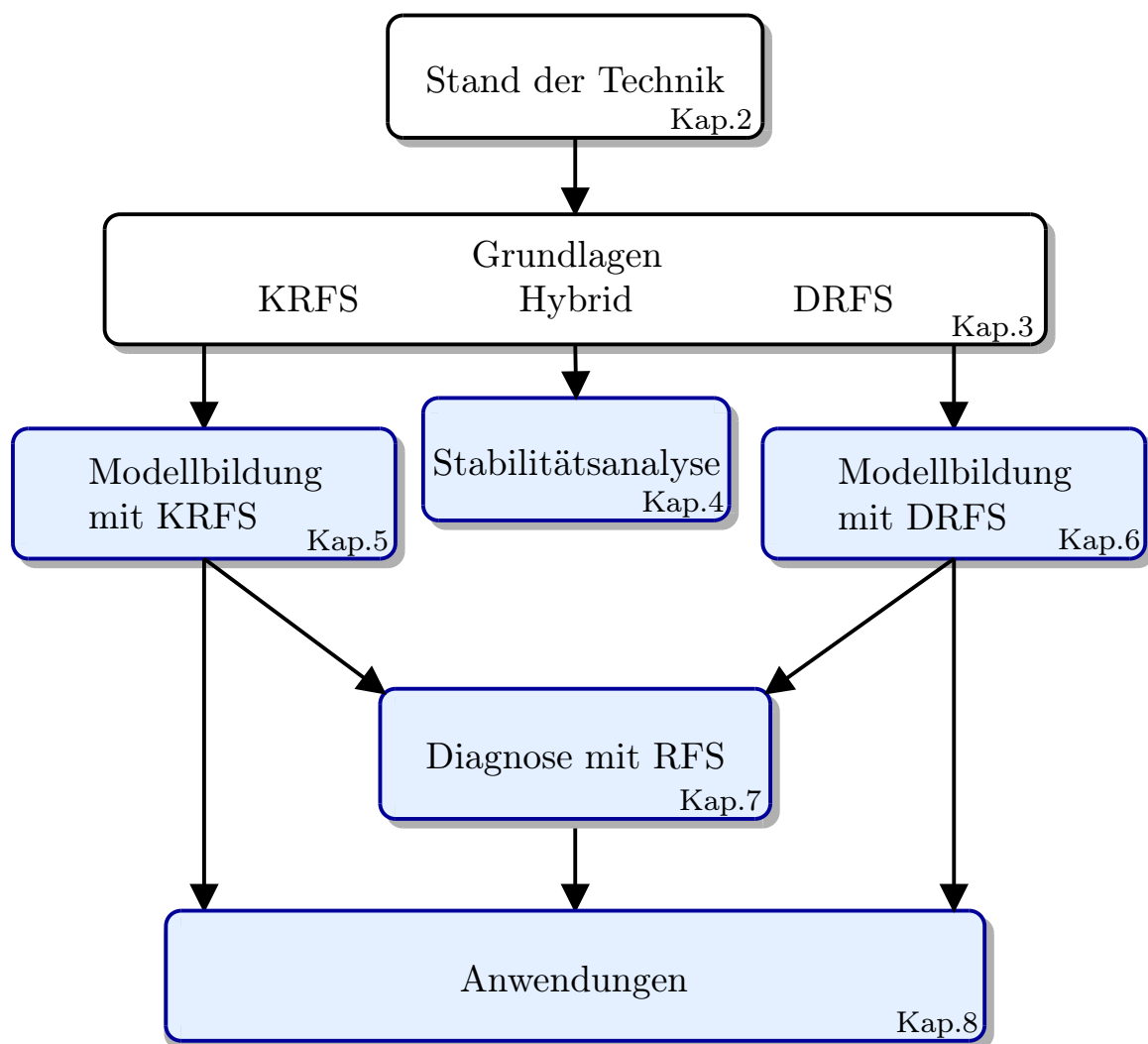


Abbildung 1.1: Übersicht über den Aufbau der Arbeit und neue Erkenntnisse (blau gekennzeichnet).

rend zeitdiskrete rekurrente Fuzzy-Systeme vor allem zur Fehlerisolation eingesetzt werden.

Mit der konkreten Anwendung der Verfahren an realen Systemen beschäftigt sich Kapitel 8. Zunächst wird die Anwendung von rekurrenten Fuzzy-Systemen zur Modellierung komplexer Systeme anhand der Nachbildung des dynamischen Verhaltens einer Destillationskolonne gezeigt. Anschließend wird die Fehlerdiagnose mithilfe zeitdiskreter rekurrenter Fuzzy-Systeme an einer hydraulischen Servoachse mit Duplex-Ventilsystem untersucht. Die Arbeit schließt mit einer Zusammenfassung in Kapitel 9.

2 Modellbildung mit Fuzzy-Systemen: ein Überblick

Seit ihrer Einführung hat sich das Gebiet der Fuzzy-Systeme kontinuierlich erweitert. Eine vielbeachtete Anwendungsmöglichkeit der Fuzzy-Logik, insbesondere bei technischen Systemen, ist die Modellierung statischer und dynamischer Zusammenhänge. Dieses Kapitel soll einen Überblick über den Stand der Technik im Bereich der Modellbildung dynamischer Systeme mit Fuzzy-Systemen geben. Hierzu werden zunächst die zur Modellbildung gebräuchlichsten Modellstrukturen von Fuzzy-Systemen vorgestellt. Daran anschließend wird auf Verfahren zur datengestützten Optimierung von Fuzzy-Systemen eingegangen. Abschließend werden weitergehende Aspekte, wie die Einbindung von Vorwissen und der Erhalt der Interpretierbarkeit der Fuzzy-Systeme, diskutiert.

2.1 Fuzzy-Modellstrukturen

Die Grundidee der Fuzzy-Logik besteht darin, menschliches Entscheidungsverhalten zu modellieren und zu mathematisieren. Analysiert man dieses Entscheidungsverhalten, so treten häufig unscharfe Übergänge zwischen charakteristischen Begriffen auf. Betrachtet man als Beispiel die Raumtemperatur, so würde ein Mensch eine Temperatur von 15° eindeutig als „zu niedrig“ und eine Temperatur von 20° als „normal“ bezeichnen. Eine Temperatur von 18° ist dagegen nicht eindeutig „zu niedrig“ oder „normal“. In einem solchen Fall ist also die *scharfe* Zuordnung von Temperaturwerten zu linguistischen Ausdrücken, wie sie Abbildung 2.1a) zeigt, nicht geeignet. Sinnvoller ist die Definition von *unscharfen* Übergängen (Abbildung 2.1b)).

Der obige Zusammenhang zwischen den Temperaturwerten und linguistischen Ausdrücken lässt sich mithilfe einer *Zugehörigkeitsfunktion* mathematisieren. Im Falle einer scharfen Zuordnung ist die Zugehörigkeitsfunktio-

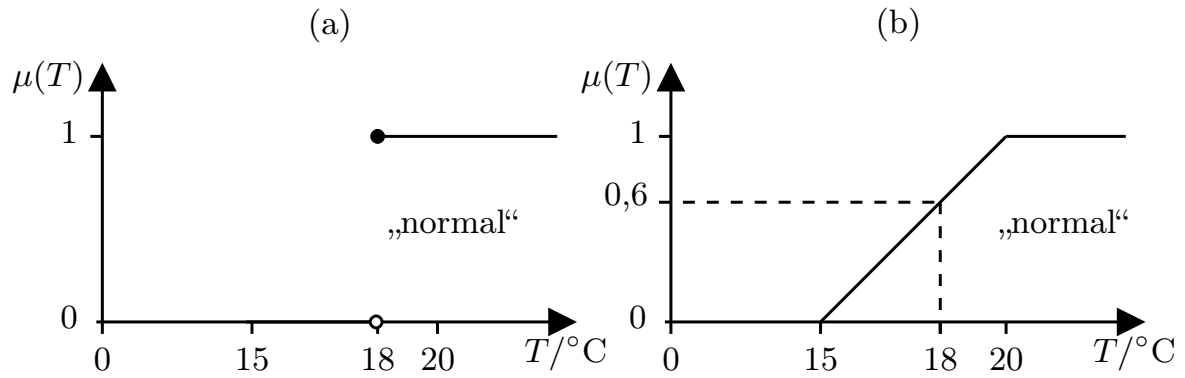


Abbildung 2.1: (a) Klassische Zugehörigkeitsfunktion (b) Fuzzy-Zugehörigkeitsfunktion.

on eine Abbildung $\mathbb{R} \rightarrow \{0,1\}$, d.h., die Raumtemperatur ist entweder „zu niedrig“ oder „normal“. Wird dagegen eine unscharfe Zuordnung gewählt, ergibt sich eine Abbildung $\mathbb{R} \rightarrow [0,1]$, die eine Zugehörigkeit zum linguistischen Ausdruck „normal“ definiert. Gemäß Abbildung 2.1b) ergibt sich somit für eine Raumtemperatur von 18° eine Zugehörigkeit von 0,6 zum linguistischen Wert „normal“. Eine solche Zuordnung eines reellen Wertes zu einem Zugehörigkeitswert wird als *Fuzzifizierung* bezeichnet.

Betrachtet man nun ausgehend von obigem Beispiel eine einfache Heizungsregelung, bei der neben der Raumtemperatur auch die Außentemperatur berücksichtigt wird, so ergibt sich ein System mit zwei unscharfen Eingangsgrößen und der Ausgangsgröße Heizung ein/aus. Für eine solche Regelung kann man beispielsweise folgende Regeln aufstellen:

- Wenn die Raumtemperatur „zu niedrig“ ist und die Außentemperatur „gering“ ist, dann schalte die „Heizung ein“.
- Wenn die Raumtemperatur „normal“ ist und die Außentemperatur „hoch“ ist, dann schalte die „Heizung aus“.
- (2.1)

Offensichtlich bilden die Regeln die Fuzzy-Mengen der Eingangsgröße Raumtemperatur \mathcal{X}_1 und der Eingangsgröße Außentemperatur \mathcal{X}_2 auf Fuzzy-Mengen der Ausgangsgröße Heizung \mathcal{Y} ab, d.h., es gilt formal $\mathcal{X}_1 \times \mathcal{X}_2 \rightarrow \mathcal{Y}$. Hierbei werden die Mengen durch Zugehörigkeitsfunktionen $\mu_{\mathcal{X}_i}$, $\mu_{\mathcal{Y}}$ beschrieben.

Die mathematische Auswertung der durch die Regeln beschriebenen Abbildung wird als *Inferenz* bezeichnet und erfolgt in mehreren Schritten [112]. Zunächst wird die Und-Verknüpfung in der Prämisse der Regel und die Konklusion auf die Ausgangsmenge mithilfe eines Und-Operators

realisiert. Ein typisches Beispiel ist der Minimum-Operator. Da meist mehrere Regeln über das System vorliegen, die jeweils mit einer bestimmten Zugehörigkeit gültig sind, müssen diese schließlich akkumuliert werden. Hierzu wird ein Oder-Operator verwendet. Ein typisches Beispiel ist hier der Maximum-Operator. Als Ergebnis der Inferenz ergibt sich so die Fuzzy-Menge der Ausgangsgröße. Da man bei vielen Systemen, wie auch bei der Heizungsregelung, an einem konkreten Zahlenwert anstatt einer Fuzzy-Menge als Ausgangsgröße interessiert ist, wird abschließend die *Defuzzifizierung* durchgeführt, die die Fuzzy-Ausgangsmenge in einen konkreten Zahlenwert umsetzt. Eine gängige Möglichkeit hierzu ist die Berechnung des Schwerpunkts der Fuzzy-Menge.

Wie bereits angedeutet, treten zumeist mehrere Eingangsgrößen auf, denen wiederum mehrere linguistische Werte zugeordnet werden. In diesem Fall bilden die Zugehörigkeitsfunktionen eine *Partitionierung des Eingangsraums*. Diese Partitionierung kann, abhängig von der Definition der Regelbasis und der Zugehörigkeitsfunktion, unterschiedliche Ausprägungen annehmen. Abbildung 2.2 zeigt die beiden gängigsten Partitionierungen für ein System mit zwei Eingängen. In Abbildung 2.2 (a) ist eine gitterähnliche Partitionierung zu sehen, wie sie bei Verwendung von abschnittsweise definierten, eindimensionalen Zugehörigkeitsfunktionen entsteht. Im Gegensatz dazu zeigt Abbildung 2.2 (b) eine Streupartitionierung, wie sie sich bei mehrdimensionalen Zugehörigkeitsfunktionen (beispielsweise Gauß-Glocken) ergibt. Durch Projektion der Gauß-Glocken auf die Koordinatenachsen können dann eindimensionale Zugehörigkeitsfunktionen erzeugt werden, die sich im Regelfall überlagern.

Neben der Partitionierung des Eingangsraums gibt es noch eine weitere wichtige Eigenschaft von Fuzzy-Systemen. Dies ist ihre linguistische Interpretierbarkeit, also die Möglichkeit, die erhaltenen Modelle linguistisch auszuwerten und damit zu plausibilisieren. Zur Bewertung der Interpretierbarkeit von Fuzzy-Systemen wurden in der Literatur verschiedenste Kriterien vorgestellt. Zhou und Gan [217] teilen diese Kriterien in zwei Klassen ein. Dies sind Kriterien zur *Low-Level*-Interpretierbarkeit, die sich auf die Definition der Zugehörigkeitsfunktionen beziehen, und Kriterien zur *High-Level*-Interpretierbarkeit, die sich auf die Definition der gesamten Regelbasis beziehen.

Zu den Low-Level-Kriterien zählen beispielsweise eine niedrige Zahl an Zugehörigkeitsfunktionen pro Dimension, für die als Richtschnur etwa 7 ± 2 gilt, sowie die Überdeckung des Definitionsbereichs der Eingangsvariablen durch Zugehörigkeitsfunktionen. Des Weiteren sollen die Zugehörigkeitsfunktionen unterscheidbar sein, sie sollten sich also nicht stark überlagern,

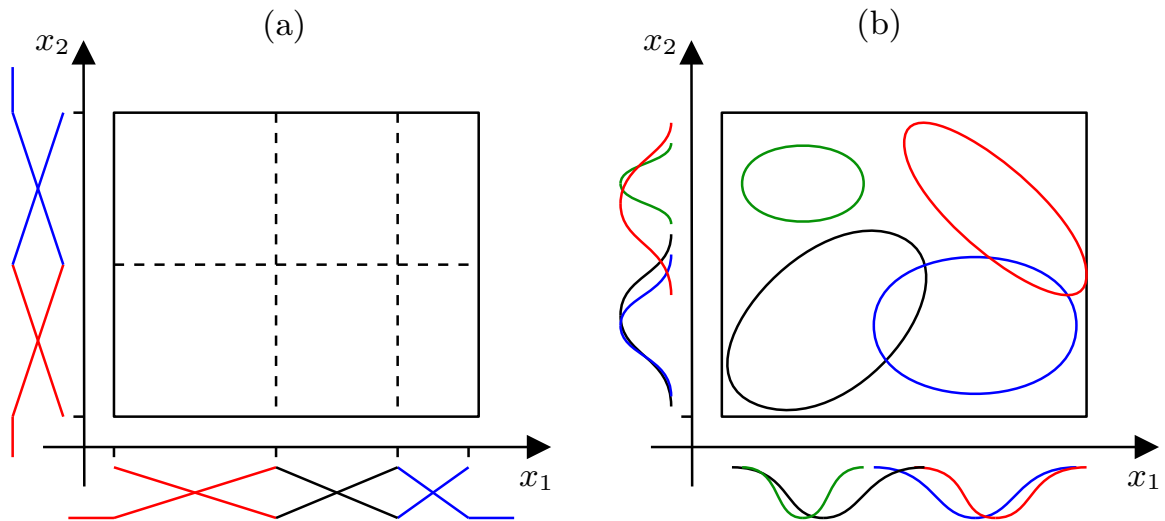


Abbildung 2.2: (a) Gitterähnliche Partitionierung. (b) Streu- bzw. prototypbasierte Partitionierung.

und sie sollen normalisiert und komplementär sein. Dementsprechend sollte die Zugehörigkeitsfunktion an mindestens einem Punkt den Maximalwert eins annehmen und die Zugehörigkeitsfunktionen sollten in jedem Punkt in der Summe eins ergeben. In [145, 146] wurden die obigen Kriterien auch durch mathematische Ausdrücke beschrieben, die den Grad an Interpretierbarkeit quantifizieren. Offenbar hängt die Low-Level-Interpretierbarkeit sehr eng mit der Partitionierung des Eingangsraums zusammen, da letztere durch die Zugehörigkeitsfunktionen bestimmt wird. Entsprechend kann man durch eine sinnvolle Definition der Partitionierung bereits eine gute Low-Level-Interpretierbarkeit erreichen.

Zu den High-Level-Kriterien zählen die Einfachheit der Regelbasis, die zumeist an der Gesamtzahl der Regeln gemessen wird, die Konsistenz der Regeln, die sichergestellt ist, wenn Regeln mit ähnlicher Prämisse keine widersprechenden Konklusionen haben, und die Vollständigkeit der Regelbasis. Außerdem sollte die Lesbarkeit einer einzelnen Regel, die sich an der Zahl an Variablen in der Prämisse bemisst, gegeben sein.

Es ist anzumerken, dass sich die obigen Kriterien unter Umständen widersprechen. Dies gilt insbesondere für den Konflikt zwischen Gesamtzahl an Regeln und Low-Level-Interpretierbarkeit. Wählt man nämlich die Zugehörigkeitsfunktionen derart, dass sie gut interpretierbar sind, führt dies aufgrund der Kriterien zu einer gitterähnlichen Partitionierung. Diese leidet aber gleichzeitig unter dem „Fluch der Dimension“, d.h., die Zahl an

Regeln steigt exponentiell mit der Zahl an Eingangsvariablen, führt also schnell auf große Regelbasen. Ansätze, diese Kriterien im Rahmen der Modellierung zu berücksichtigen, werden in Abschnitt 2.2.4 beschrieben.

Fuzzy-Systeme, die über Regeln der Form (2.1) definiert werden, können nur statisches Verhalten nachbilden. Will man auch dynamische Zusammenhänge über Fuzzy-Systeme modellieren, so muss man dynamische Elemente in die Systemstruktur integrieren, was auf dynamische Fuzzy-Systeme führt.

Im Folgenden sollen die zur Modellierung gebräuchlichsten Arten dynamischer Fuzzy-Systeme, namentlich linguistische Fuzzy-Systeme nach Mamdani (MFS) [130], Takagi-Sugeno-Kang (TSK) Fuzzy-Systeme [187] und die verwandten Neuro-Fuzzy-Systeme (NFS) sowie rekurrente Fuzzy-Systeme (RFS) [5, 65] vorgestellt und verglichen werden. Weitere Strukturmodelle, wie Fuzzy-Relational-Modelle [153] oder Typ-2 Fuzzy-Systeme [96, 213], werden aufgrund ihrer bisher geringen praktischen Bedeutung nicht weiter betrachtet.

Mamdani Fuzzy-Systeme

Eine erste Anwendung der Theorie der Fuzzy-Mengen zur Modellbildung und später auch zur Regelung wurde 1977 durch Mamdani [130] in Form von linguistischen, regelbasierten Fuzzy-Systemen vorgestellt. Die Regeln nehmen dabei die Form

$$\text{Wenn } \mathbf{z} = \mathbf{L}_j^{\mathbf{z}}, \quad \text{dann } \mathbf{y} = \mathbf{L}_v^{\mathbf{y}}$$

an, wobei \mathbf{z} den Vektor der Eingangsgrößen und \mathbf{y} den Vektor der Ausgangsgrößen bezeichnet. Die Komponenten $L_{j_i}^{z_i}$ bzw. $L_{v_r}^{y_r}$ der linguistischen Vektoren $\mathbf{L}_j^{\mathbf{z}}$ bzw. $\mathbf{L}_v^{\mathbf{y}}$ stellen die linguistischen Werte der i -ten Eingangs- bzw. r -ten Ausgangsvariablen dar. Als Operatoren der Inferenz werden der Minimum- und Maximum-Operator, als Zugehörigkeitsfunktionen zumeist Dreieck- und Trapezfunktionen verwendet. Durch diese Wahl ergibt sich eine gitterähnliche Partitionierung des Eingangsraums.

Die Dynamik der MFS wird durch äußere Rückführung der Ausgangsgrößen erzeugt. In diesem Fall setzt sich der Vektor \mathbf{z} gemäß

$$\mathbf{z} = [u(k) \ u(k-1) \ \dots \ u(k-v) \ y(k-1) \ \dots \ y(k-w)] \quad (2.2)$$

aus den zeitverzögerten Eingängen $u(k-i)$ und Ausgängen $y(k-i)$ zusammen. Aufgrund der guten Interpretierbarkeit wird die Modellbildung bei MFS zumeist wissensbasiert durchgeführt, d.h., die Regeln werden durch Expertenwissen festgelegt.

Takagi-Sugeno-Kang Fuzzy-Systeme

Eine weitere Systemstruktur sind TSK Fuzzy-Systeme, die seit ihrer Einführung im Jahr 1985 in [187] große Verbreitung zur Modellbildung dynamischer Systeme gefunden haben. Prinzipiell existieren zwei Ausprägungen: TSK-Systeme in Ein-/Ausgangsdarstellung und TSK-Systeme in Zustandsraumdarstellung. Für die Ein-/Ausgangsdarstellung lassen sich die folgenden Regeln angeben:

$$\text{Wenn } \mathbf{z} = \mathbf{L}_j^{\mathbf{z}}, \quad \text{dann } y = \mathbf{a}_i^T \mathbf{z} + b_i, \quad i = 1, \dots, K, \quad (2.3)$$

wobei $\mathbf{L}_j^{\mathbf{z}}$ wiederum der linguistische Vektor der Eingangsgrößen \mathbf{z} gemäß Gl. (2.2) ist, K die Zahl an Regeln angibt und die Parameter \mathbf{a}_i^T , b_i eine affin-lineare Funktion definieren. Im MIMO-Fall werden die Ausgänge zumeist unabhängig voneinander betrachtet, d.h., es treten m Gleichungen der Form (2.3) auf. Als Fuzzy-Operatoren für Aggregation und Implikation bzw. Akkumulation werden Produkt bzw. einfache Summe verwendet.

Im Gegensatz zu MFS ist die Konklusion bei TSK Fuzzy-Systemen nicht mehr durch linguistische Größen, sondern durch eine affin-lineare Funktion beschrieben. Entsprechend sind TSK Fuzzy-Systeme nicht mehr linguistisch interpretierbar. Vielmehr lässt sich das TSK Fuzzy-System als eine Überlagerung von affin-linearen Teilmodellen interpretieren. Wie bei MFS wird die Dynamik durch die Wahl des Eingangsgrößenvektors \mathbf{x} , d.h. durch Rückführung der Ausgangsgrößen, erzeugt.

Für die Wahl der Zugehörigkeitsfunktionen existieren verschiedene Möglichkeiten. Zum einen kann man wieder eindimensionale Zugehörigkeitsfunktionen wählen, was auf eine gitterähnliche Partitionierung führt. Zum anderen kann man mehrdimensionale Zugehörigkeitsfunktionen, z.B. Gauß-Glocken, wählen, sodass sich eine Streupartitionierung ergibt.

Neben der oben angegebenen Ein-/Ausgangsbeschreibung kann man TSK Fuzzy-Systeme auch als Zustandsraummodelle definieren. Dann ergeben sich Regeln der Form

$$\begin{aligned} \text{Wenn } \mathbf{x} = \mathbf{L}_j^{\mathbf{x}} \text{ und } \mathbf{u} = \mathbf{L}_q^{\mathbf{u}}, \\ \text{dann } \delta \mathbf{x} = \mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u} \text{ und } \mathbf{y} = \mathbf{C}_i \mathbf{x} + \mathbf{D}_i \mathbf{u}, \quad i = 1, \dots, K, \end{aligned}$$

wobei $\delta \mathbf{x} = \dot{\mathbf{x}}$ im zeitkontinuierlichen und $\delta \mathbf{x} = \mathbf{x}(k+1)$ im zeitdiskreten Fall gilt. So definierte TSK Fuzzy-Systeme kann man als Überlagerung lokaler Zustandsraummodelle, die im i -ten Arbeitspunkt linearisiert wurden, interpretieren. Wird nicht in der Ruhelage linearisiert, tritt zusätzlich

ein affiner Term \mathbf{c}_i in der Zustandsgleichung auf. Solche lokal-affinen TSK-Modelle im Zustandsraum werden in [110] behandelt.

Im Gegensatz zu Ein-/Ausgangsmodellen, die meist datengestützt erzeugt werden, werden TSK-Modelle im Zustandsraum meist durch Linearisierung nichtlinearer Systemmodelle gebildet, die durch physikalische Modellbildung entstanden sind. Ein bekanntes Verfahren ist der *Sector-Nonlinearity-Ansatz* nach Tanaka und Wang [189]. Weitere Verfahren finden sich u.a. in [87, 88, 188].

Neuro-Fuzzy-Systeme

Die Grundidee bei der Entwicklung von Neuro-Fuzzy-Systemen liegt darin, die guten Lernfähigkeiten neuronaler Netze mit der Interpretierbarkeit von Fuzzy-Systemen zu verknüpfen. Dies wird dadurch ermöglicht, dass einige Arten von künstlichen neuronalen Netzen, insbesondere radiale Basisfunktionennetze, einen strukturell nahezu gleichen Aufbau haben wie TSK Fuzzy-Systeme in Ein-/Ausgangsdarstellung. Entsprechend stellt man das Fuzzy-System als neuronales Netz dar, dessen Schichten die Schritte Fuzzifizierung, Inferenz und Defuzzifizierung repräsentieren. Abbildung 2.3 zeigt die NFS Darstellung eines TSK Fuzzy-Systems, das Ausgangspunkt für verschiedene strukturelle Weiterentwicklungen war. Dies betrifft vor allem die Gestaltung der Rückführung, mit der dynamisches Verhalten eingebracht wird. Es lassen sich globale und lokale Rückführungen unterscheiden, wie Abbildung 2.3 zeigt. Bei den mit TSK Fuzzy-Systemen identischen NFS werden die Ausgänge eines Feed-Forward-Netzes zeitverzögert als zusätzliche Eingänge zurückgeführt (strichpunktiert). Alternativ existieren NFS, bei dem die Rückführung nach der Regelprämisse erfolgt (strichliert) [95, 192, 216]. Bei beiden Varianten erfolgt die Rückführung über mehrere Schichten, weshalb sie als globale Rückführungen bezeichnet werden. Dagegen wird bei den in [114, 115, 132] eingeführten NFS die Rückführung direkt am Neuron vorgenommen (vgl. Abbildung 2.3, durchgezogene Linie), weshalb man von lokaler Rückführung spricht. Als Zugehörigkeitsfunktionen werden im Regelfall Gauß- und Sigmoid-Funktionen verwendet. Ihrer Grundidee entsprechend erfolgt die Modellierung ausschließlich datengestützt. Abschließend sei angemerkt, dass alle entwickelten NFS jeweils Ein-/Ausgangsverhalten nachbilden. NFS, bei denen auch interne Zusammenhänge über Zustandsgrößen abgebildet werden, gibt es dagegen bisher nicht.

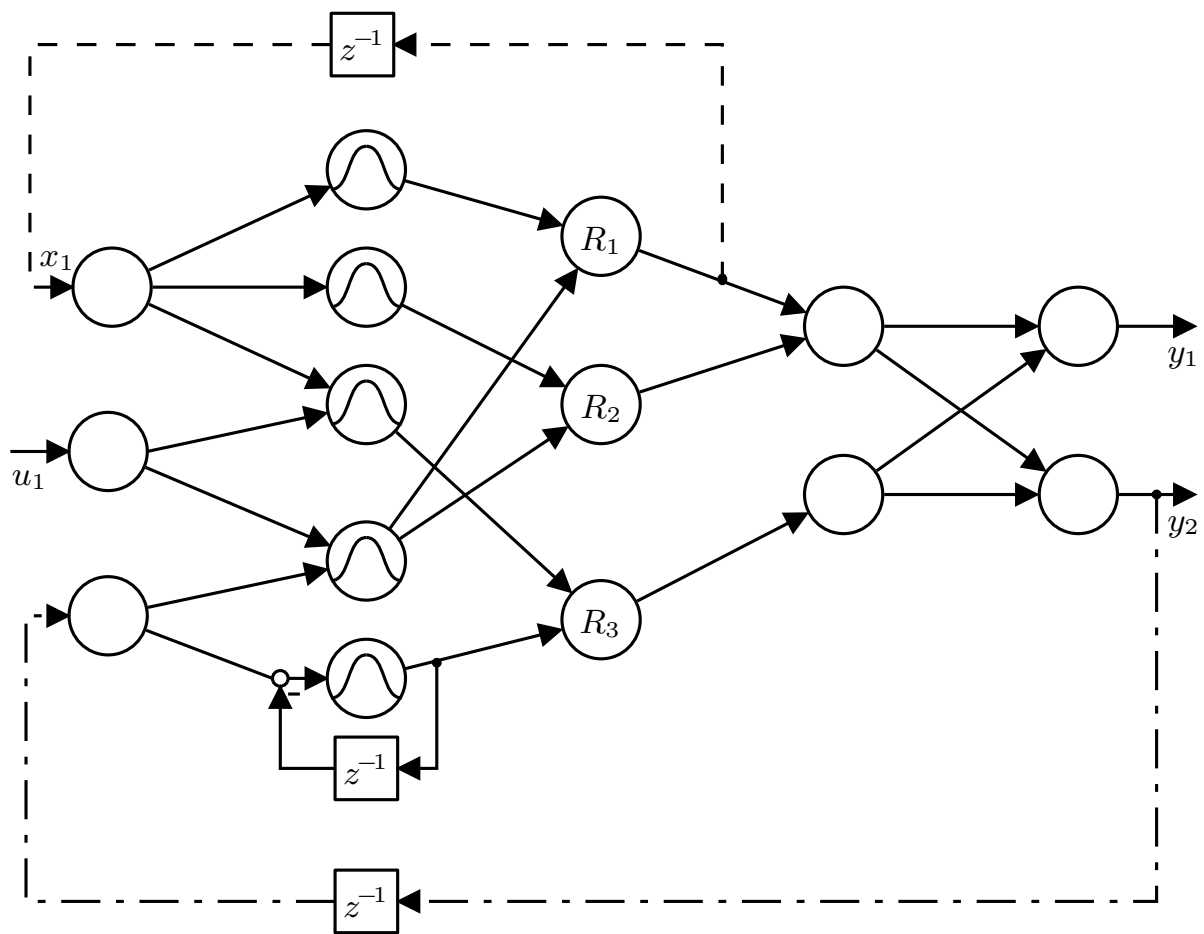


Abbildung 2.3: Aufbau eines Neuro-Fuzzy-Systems mit Rückführung des Ausgangs (strichpunktiert), Rückführung nach der Regelprämisse (strichliert) und lokaler Rückführung am Neuron (durchgezogen).

Rekurrente Fuzzy-Systeme

Eine weitere Modellstruktur sind rekurrente Fuzzy-Systeme, die zunächst im Zeitdiskreten in [2] und [65] unabhängig voneinander und aus unterschiedlichen Motiven heraus entwickelt wurden. In [2] lag der Fokus auf der Verbindung von zeitdiskreten rekurrenten Fuzzy-Systemen (DRFS) zu Automaten, während [65] die Verbindung DRFS zu rekurrenten neuronalen Netzen untersuchte. Auf Basis dieser Arbeiten wurden DRFS in [5, 98] formalisiert und systemtheoretisch untersucht. In [4, 49] wurden RFS dann auf den Fall zeitkontinuierlicher Größen erweitert. Prinzipiell handelt es sich um nichtlineare linguistische Zustandsraummodelle, bei de-

nen die Dynamik- und Ausgangsfunktion über Regeln der Form

$$\begin{aligned} \text{Wenn } \mathbf{x} &= \mathbf{L}_j^{\mathbf{x}} \quad \text{und} \quad \mathbf{u} = \mathbf{L}_q^{\mathbf{u}}, \\ \text{dann } \delta\mathbf{x} &= \mathbf{L}_w^{\delta\mathbf{x}} \quad \text{und} \quad \mathbf{y} = \mathbf{L}_v^{\mathbf{y}} \end{aligned}$$

bestimmt werden, wobei wiederum $\delta\mathbf{x} = \dot{\mathbf{x}}$ im zeitkontinuierlichen und $\delta\mathbf{x} = \mathbf{x}(k+1)$ im zeitdiskreten Fall gilt. Als Fuzzy-Operatoren werden wie bei TSK Fuzzy-Systemen Produkt und einfache Summe verwendet. Als Zugehörigkeitsfunktionen wählt man Dreieck- und Rampenfunktionen für \mathbf{x} und \mathbf{u} sowie Singletons für $\delta\mathbf{x}$ und \mathbf{y} .

Bislang wurde die Modellierung, insbesondere aufgrund der guten linguistischen Interpretierbarkeit, weitestgehend wissensbasiert durchgeführt, wie in [49] am Beispiel einer Brennstoffzelle gezeigt wird. Die datengestützte Modellierung wird im Rahmen dieser Arbeit betrachtet.

Vergleich der Modellstrukturen

Die zuvor vorgestellten Modellstrukturen sollen abschließend anhand der Kriterien Systemdarstellung, Art der Modellbildung¹, Partitionierung des Eingangsraums und linguistische Interpretierbarkeit verglichen werden. Einen Überblick gibt Tabelle 2.1.

Tabelle 2.1: Vergleich der Modellstrukturen linguistische Fuzzy-Systemen nach Mamdani (MFS), TSK Fuzzy-Systeme in Ein-/Ausgangsdarstellung (EA-TSK), TSK Fuzzy-Systeme in Zustandsraumdarstellung (ZR-TSK), Neuro-Fuzzy-Systeme (NFS) und rekurrente Fuzzy-Systeme (RFS). Die Modellbildung erfolgt entweder wissensbasiert (wb), datengestützt (dg) oder physikalisch (phy).

	MFS	EA-TSK	ZR-TSK	NFS	RFS
Systemdarstellung	EA	EA	ZR	EA	ZR
Art der Modellierung	wb	dg	phy	dg	wb
Partitionierung	Gitter	Streu	Gitter	Streu	Gitter
ling. interpretierbar	+	—	—	—	+

Linguistische Fuzzy-Systeme nach Mamdani und Neuro-Fuzzy-Systeme bilden ausschließlich Ein-/Ausgangsverhalten nach, während es sich bei

¹Hier wird jeweils auf den in der Literatur dominierenden Typ bzw. die historische Motivation hingewiesen. Teilweise existieren auch alternative Ansätze mit geringer Verbreitung.

rekurrenten Fuzzy-Systemen um regelbasierte Zustandsraummodelle handelt. Ein Sonderfall sind TSK Fuzzy-Systeme. Für sie gibt es sowohl eine Ein-/Ausgangsdarstellung als auch eine Zustandsraumdarstellung.

Die Modellbildung erfolgt bei MFS zumeist wissensbasiert, es wird also qualitatives Prozesswissen in Regeln umgesetzt. Dies entspricht auch der ursprünglichen Idee hinter der Entwicklung von MFS. Es gibt allerdings auch Ansätze, MFS datengestützt zu entwerfen oder zumindest zu optimieren [32, 81, 107]. EA-TSK Systeme und NFS werden dagegen fast ausschließlich datengestützt entworfen, während ZR-TSK Systeme im Regelfall aus der physikalischen Modellbeschreibung nichtlinearer Systeme abgeleitet werden. Alternativ können ZR-TSK Modelle auch aus der zugehörigen Ein-/Ausgangsbeschreibung abgeleitet werden [111]. RFS werden bisher meist wissensbasiert entworfen. Die datengestützte Modellbildung wird in dieser Arbeit entwickelt.

Die Partitionierung des Eingangsraums ist bei MFS und RFS gitterähnlich. Dagegen spricht man bei der Partitionierung von datengestützt entworfenen TSK Systemen bei Verwendung von mehrdimensionalen Zugehörigkeitsfunktion von einer streu- oder prototypbasierten Partitionierung. Zumeist ergeben sich elliptische Höhenlinien. Dies gilt allerdings nur, wenn keine Projektionen auf die Achsen der Eingangsgrößen durchgeführt werden. Bei ZR-TSK Modellen werden oft auch Gitterpartitionierungen verwendet. Bei den meisten Formen von NFS gibt es dagegen aufgrund der Rückführungen in den Inferenz-Schritten keine direkte Partitionierung.

Eng verbunden mit der Partitionierung ist die linguistische Interpretierbarkeit. Ist die Partitionierung gitterähnlich, so lassen sich direkt Regeln ableiten. Entsprechend ist die linguistische Interpretierbarkeit bei MFS und RFS sehr gut. Bei TSK Modellen dagegen ist die linguistische Interpretierbarkeit kaum gegeben, da in der Konklusion keine linguistischen Werte, sondern lineare Funktionen angegeben werden. NFS sind zumeist reine Funktionsapproximatoren, sodass die linguistische Interpretierbarkeit fast vollständig verloren geht.

2.2 Modellbildung mit Fuzzy-Systemen

Nachdem im vorigen Abschnitt die Modellstrukturen vorgestellt wurden, soll nun ein Überblick über verschiedene Verfahren gegeben werden, mit denen dynamische Systeme durch dynamische Fuzzy-Systeme nachgebildet werden können.

2.2.1 Wissensbasierte und datengestützte Modellbildung

Die ursprüngliche Idee, Fuzzy-Logik zur Modellierung von Systemen einzusetzen, war dadurch motiviert, bestehendes Wissen von Experten in einem spezifischen Anwendungsbereich nutzbar zu machen. Entsprechend zeichnen sich die ersten Ansätze zur Modellbildung dadurch aus, das bestehende Kausalwissen über den zu modellierenden Prozess in Form von Regeln darzustellen und schließlich mithilfe der Fuzzy-Logik zu mathematisieren [130]. Hierzu wurden zumeist Mamdani Fuzzy-Systeme verwendet. Allerdings hat dieser Ansatz Grenzen. Zum einen ist das Wissen bei vielen Prozessen nicht ausreichend vorhanden, zum anderen ist mithilfe von Kausalwissen nur eine qualitative Modellbildung möglich. Die Quantifizierung des Modells allein aufgrund von Prozesswissen ist dagegen schwierig.

Aus diesen Gründen entwickelte sich die Forschung zunehmend in Richtung datengestützter Verfahren, mit denen die qualitativen Modelle quantifiziert werden können. Insbesondere die Einführung der TSK-Fuzzy-Systeme [187] im Jahr 1985 lenkte die Forschungsaktivitäten hin zu einer ausschließlich auf Messdaten basierenden Modellbildung. In der Folge wurde eine Vielzahl von Ansätzen entwickelt. Die Entwicklung der Neuro-Fuzzy-Systeme ermöglichte des Weiteren die Anwendung von Modellierungsverfahren für neuronale Netze auf Fuzzy-Systeme. Die zunehmend datengestützte Modellierung führte allerdings dazu, dass Fuzzy-Systeme, ähnlich wie neuronale Netze, vornehmlich als universelle Approximatoren eingesetzt wurden. Die Verwendung von Prozesswissen sowie die Interpretierbarkeit der erhaltenen Modelle rückte in den Hintergrund, wodurch die ursprüngliche Idee der Fuzzy-Modellbildung teilweise verloren ging. Dieser Entwicklung wurde in jüngerer Zeit durch Verfahren, die sich mit dem Einbringen von Vorwissen und dem Erhalt der Interpretierbarkeit bei gleichzeitig guten Approximationseigenschaften beschäftigen, entgegen gewirkt [12, 29, 67, 86].

Es sollen nun zunächst die wichtigsten Verfahren zur Lösung der beiden Kernprobleme der datengestützten Modellbildung, der Struktur- und der Parameteroptimierung, diskutiert werden. Abschließend wird auf den Erhalt der Interpretierbarkeit eingegangen.

2.2.2 Strukturoptimierung

Unter der Optimierung der Struktur versteht man insbesondere die Festlegung der Zahl an Zugehörigkeitsfunktionen für die Eingangsgrößen sowie deren Gültigkeitsbereich. Dies wird häufig auch als Partitionierung des Ein-

gangsraums bezeichnet. In der Literatur sind hierzu viele verschiedene Ansätze zu finden, die sich grob in iterativ-heuristische Verfahren, Clustering und Metaheuristiken einteilen lassen. Im Folgenden werden die Vorgehensweisen kurz erläutert.

Iterativ-heuristische Verfahren

Der erste Ansatz zur iterativen Konstruktion von Fuzzy-Systemen wurde von Wang und Mendel in [204] vorgestellt und besteht darin, jedem Trainingsdatenpaar eine Regel zuzuweisen. Dieser sehr einfache Ansatz führt allerdings bei größeren Datensätzen zu einer nicht mehr überschaubaren Zahl an Regeln. Entsprechend wurden verschiedene Erweiterungen eingeführt, die weiterhin die Datenpaare einzeln verarbeiten, gleichzeitig aber die Regelanzahl begrenzen. So ergänzen Nauck und Kruse [141] nur dann eine Regel, wenn das Datenpaar durch die bestehende Regelbasis nicht genügend abgebildet ist. Dies wird über ein Distanzmaß berechnet. Ähnliche Verfahren finden sich u.a. in [143]. Möglich ist auch, die Regelauswahl mithilfe statistischer Test durchzuführen [107, 181]. Dagegen arbeiten [44, 140, 203] mit Orthogonalisierungsmethoden wie der orthogonalen Least-Squares-Schätzung (OLS) oder einer Singulärwertzerlegung, um die Relevanz der durch die Datenpaare erzeugten Regeln zu bestimmen. Alle nicht relevanten Regeln werden schließlich entfernt. Weitere hierzu verwandte Verfahren stellen [132, 211] vor.

Eine weitere Gruppe sind baumstrukturbasierte Heuristiken [27, 83, 117, 142, 157]. Die Grundidee dieser Verfahren besteht darin, mit einer einfachen Anfangspartitionierung (oftmals nur eine Zugehörigkeitsfunktion pro Eingang) zu starten und dem Fuzzy-System schrittweise neue Zugehörigkeitsfunktionen hinzuzufügen. Ein bekannter Algorithmus ist *Lolimot* [142], bei dem in einem Iterationsschritt nacheinander jeder Dimension eine Zugehörigkeitsfunktion hinzugefügt und der entstehende Gütewert berechnet wird. Die Zugehörigkeitsfunktion, die den besten Gütewert erzeugt, wird schließlich hinzugefügt. Auf diese Weise wird die Partitionierung auf Basis des im vorigen Schritt ermittelten Modells verfeinert, sodass eine Baumstruktur entsteht. Als Gütewert wird in *Lolimot* der quadratische Prädiktionsfehler verwendet. Alternativ können auch Wahrscheinlichkeitsmaße verwendet werden [83]. Neben den Gütewerten unterscheiden sich die Algorithmen auch in der Bestimmung der Position im Eingangsraum, an der neue Zugehörigkeitsfunktionen eingefügt werden. In [117, 142] wird zunächst der Bereich des Eingangsraums bestimmt, in dem der Prädiktionsfehler am größten ist. Anschließend wird dieses Gebiet mit einem festen Verhältnis

unterteilt. Dagegen wird in [27, 157] die Position der neuen Zugehörigkeitsfunktion abhängig vom Prädiktionsfehler festgelegt.

Clustering

Ein zweiter Ansatz zur Strukturoptimierung sind Clustering-Verfahren, die streng genommen auch zu den Heuristiken gezählt werden können, aufgrund ihrer großen Bedeutung aber gesondert vorgestellt werden. Grundsätzlich kann man zwei Arten von Clustering-Algorithmen unterscheiden: Offline- und Online-Algorithmen. Erstere basieren auf einem vollständigen Datensatz des Systems, letztere dagegen starten mit einem Datenpaar und passen dann das bestehende System an neue Datenpaare an.

Als Offline-Ansatz wird stellvertretend das Verfahren nach Kroll [108] vorgestellt werden. Hierbei wird auf den durch die Datensätze der Ein- und Ausgangsgrößen aufgespannten Raum eines der bekannten Fuzzy-Clusteringverfahren, beispielsweise der Fuzzy-C-Means Algorithmus, angewendet. Dadurch werden Datenpaare, die im Sinne eines Distanzmaßes nahe beieinander liegen, in einen gemeinsamen Cluster zusammengefasst. Hierbei muss die Gesamtzahl der Cluster vorher festgelegt werden. Da beim Fuzzy-C-Means mit jedem Cluster eine mehrdimensionale Zugehörigkeitsfunktion verbunden ist, kann diese direkt verwendet werden. Möchte man eindimensionale Zugehörigkeitsfunktionen für jede Ein- und Ausgangsgröße erhalten, so werden die mehrdimensionalen Zugehörigkeitsfunktionen auf die jeweiligen Koordinatenachsen projiziert. Ähnliche Ansätze finden sich u.a. in [1, 13, 30, 39, 108, 184], die sich im Wesentlichen im verwendeten Clusteringalgorithmus und in der Art der Projektionen unterscheiden.

Die Online-Ansätze [93, 115, 124, 192] betrachten dagegen in jedem Schritt nur ein Datenpaar und arbeiten wie folgt: Anhand des ersten Datenpaares wird ein initialer Cluster gebildet, dessen Mittelpunkt den Eingangskoordinaten des Datenpaares entspricht. Die Varianz des Clusters wird auf einen einstellbaren Wert festgelegt. Steht nun ein weiteres Datenpaar zur Verfügung, wird die Distanz zu allen bisherigen Clustern ermittelt. Übersteigt diese Distanz eine Grenze, wird ein neuer Cluster, wie oben beschrieben, erzeugt. Ist die Distanz geringer, wird das Datenpaar dem nächstgelegenen Cluster zugeordnet. Außerdem werden Mittelpunkt und Varianz angepasst. Auf diese Weise wird sukzessive die Struktur aufgebaut. Einige Algorithmen ermitteln zusätzlich die Relevanz und den Grad an Überlappung der Cluster, sodass einzelne, nicht mehr relevante Cluster wieder entfernt werden können. Wie auch beim Offline-Ansatz können

schließlich die höherdimensionalen Cluster auf die Eingangsgrößen projiziert werden.

Metaheuristiken

Eine dritte Gruppe von Verfahren zur Strukturoptimierung sind sogenannte Metaheuristiken, die unter anderem Genetische Algorithmen, Genetische Programmierung, Partikelschwarmoptimierung und Ameisenalgorithmen umfassen. Die Grundidee, die allen Metaheuristiken gemeinsam ist, besteht darin, eine größere Zahl an Individuen, die die Lösungen des Optimierungsproblems repräsentieren, im Suchraum zu verteilen und anschließend in Richtung des Optimums zu bewegen. Die Bewegung wird dabei teils zufällig, teils abhängig vom Gütewert gesteuert.

Die Anwendung solcher Metaheuristiken zur Strukturoptimierung von Fuzzy-Systemen erfordert nun zum einen die geeignete Codierung der Struktur im Individuum und zum anderen die Definition des Lösungsalgorithmus und seiner Parameter. Entsprechend unterscheiden sich die verschiedenen Ansätze insbesondere in diesen Punkten.

Bei genetischen Algorithmen wird die Struktur in einem Chromosom codiert. Um Regelbasen mit variierenden Größen zu ermöglichen, werden entweder Bits eingeführt, die eine Regel zu oder abschalten, oder Chromosome mit variabler Länge zugelassen [10, 33, 64, 73, 91, 163, 186]. Bei Ansätzen mit genetischer Programmierung [17, 72] wird das Fuzzy-System dagegen in einer Baumstruktur dargestellt. Variierende Größen der Regelbasis können so relativ einfach durch Einstellung der Baumknoten berücksichtigt werden [17]. Die Lösung der so definierten Optimierungsprobleme erfolgt dann in den üblichen Schritten Reproduktion, Mutation und Selektion [171]. Eine breite Übersicht über genetische Fuzzy-Systeme bietet [31].

Dagegen stellt Juang [92] die Optimierung des Fuzzy-Systems als gemischt-ganzzahliges Optimierungsproblem dar, das mithilfe eines im kontinuierlichen Suchraum arbeitenden Ameisenalgorithmus gelöst wird. Ähnliche Ansätze verwenden anstelle des kontinuierlichen Ameisenalgorithmus auch die Partikelschwarmoptimierung [90, 94]. Bei beiden Verfahren erfolgt die Optimierung durch die Nachbildung von Schwarmverhalten [100], d.h., die Optimierungsparameter werden abhängig vom eigenen bisher besten Gütewert und vom besten Gütewert des Schwarms verändert.

Vergleich der Verfahren

Abschließend sollen die Verfahren zur Strukturoptimierung bezüglich der Kriterien Partitionierung des Eingangsraums, Rechenaufwand, Zahl der Einstellparameter, Möglichkeit zur Online-Optimierung und Interpretierbarkeit der erhaltenen Modelle verglichen werden. Einen Überblick gibt Tabelle 2.2.

Tabelle 2.2: Vergleich der Modellierungsverfahren iterativ-heuristisch nach Wang und Mendel (WM-IH), baumbasierte IH (B-IH), Offline-Clustering (OFF-C), Online-Clustering (ON-C) und evolutionäre Algorithmen (Evos) nach den Kriterien Partitionierung des Eingangsraums, Rechenaufwand, Zahl und Einstellbarkeit der Parameter sowie High-Level- (HL) und Low-Level- (LL)-Interpretierbarkeit. Die Abkürzungen S und P stehen für Struktur bzw. Parameter.

	WM-IH	B-IH	OFF-C	ON-C	Evos
Partitionierung	Gitter	Gitter	Streu	Streu	frei
Rechenaufwand	+	○	+	+	—
Einstellparameter	+	+	+	+	—
Online-Optimierung	S+P	P	P	S+P	keine
HL-Interpretierbarkeit	—	—	+	+	+
LL-Interpretierbarkeit	—	+	—	—	+

Die Partitionierung ist bei den iterativ-heuristischen Methoden gitterähnlich, da sie direkt mit eindimensionalen Zugehörigkeitsfunktionen arbeiten. Dagegen führen Clusteringverfahren auf Streupartitionierungen. Bei evolutionäre Algorithmen existieren prinzipiell beide Varianten, die Partitionierung kann hier vom Anwender vorgegeben werden.

Der Rechenaufwand ist bei allen Verfahren mit Ausnahme der evolutionäre Algorithmen relativ moderat. Es werden jeweils nur wenige Rechenschritte benötigt. Allerdings skaliert die Optimierung bei baumbasierten Heuristiken mit der Zahl an Regeln, sodass sich bei höherdimensionalen Systemen der Rechenaufwand erhöht. Dies wirkt sich beim Clustering durch die Verwendung mehrdimensionaler Zugehörigkeitsfunktionen weniger stark aus.

Die Zahl der Einstellparameter ist bei evolutionäre Algorithmen relativ hoch sowie die Einstellung der Parameter oft schwierig. Es müssen

u.a. Mutations- und Rekombinationswahrscheinlichkeiten festgelegt werden, was eine gewisse Erfahrung im Umgang mit evolutionäre Algorithmen voraussetzt. Dagegen sind bei den anderen Verfahren meist nur sehr wenige Parameter, wie Abbruchkriterien und Distanzmaße einzustellen, die für den Anwender leicht verständlich sind.

Die Online-Optimierung lässt sich nochmal in Online-Optimierung der Struktur und der Parameter aufteilen. Beides ist mithilfe von Online-Clustering sowie mit der Methode nach Wang und Mendel [204] möglich. Dagegen lassen sich mit Offline-Clustering und baumbasierte Heuristiken nur die Parameter online einstellen. Die Struktur muss vorher feststehen. Mit evolutionäre Algorithmen ist aufgrund des hohen Rechenaufwands keine Online-Optimierung möglich.

Die High-Level-Interpretierbarkeit ist bei Clusteringverfahren sehr gut, da sie zumeist nur wenige Regeln mit guter Lesbarkeit generieren. Hier haben Heuristiken Nachteile, da sie insbesondere bei höherdimensionalen Systemen sehr viele Regeln erzeugen. Umgekehrt verhält es sich bei der Low-Level-Interpretierbarkeit. Hier haben insbesondere baumbasierte Heuristiken Vorteile, da die Zugehörigkeitsfunktionen sehr gut unterscheidbar sind. Dies ist bei Methode nach Wang und Mendel und beim Clustering im Allgemeinen nicht erfüllt. Metaheuristiken führen bei geeigneten Gütekriterien meist auf Regelbasen mit guter High-Level-Interpretierbarkeit. Auch die Low-Level-Interpretierbarkeit kann im Gütekriterium berücksichtigt werden (vgl. Abschnitt 2.2.4), was zu einer guten Interpretierbarkeit führt.

2.2.3 Parameteroptimierung

Unter den Parametern von Fuzzy-Systemen versteht man im Wesentlichen die Parameter der Zugehörigkeitsfunktionen bei regelbasierten Fuzzy-Systemen bzw. die Koeffizienten der linearen Teilmodelle bei TSK Fuzzy-Systemen. Bei Neuro-Fuzzy-Systemen sind die Netzgewichte die freien Parameter.

Im Gegensatz zur Strukturoptimierung ist die Optimierung dieser Parameter ein Standardproblem der Optimierung. Als Gütekriterium verwendet man im Regelfall den quadratischen Fehler zwischen gemessenen und durch das Modell nachgebildeten Daten. Dies führt auf ein kontinuierliches Optimierungsproblem, dessen Optimierungsparameter in zwei Gruppen eingeteilt werden können [13]. Eine Parametergruppe (im Folgenden *lineare Parameter* genannt) führt auf ein Optimierungsproblem, das *linear* in den Parametern ist, die andere Gruppe (im Folgenden *nichtlineare*

Parameter genannt) auf ein Optimierungsproblem, das *nichtlinear* in den Parametern ist. Beispiele für lineare Parameter sind die Koeffizienten der linearen Teilmodelle bei TSK Fuzzy-Systemen, während die Parameter der Eingangszugehörigkeitsfunktionen nichtlineare Parameter sind.

Die Optimierung der linearen Parameter lässt sich als Least-Squares (LS) Optimierungsproblem formulieren, zu dessen Lösung verschiedene Möglichkeiten zur Verfügung stehen [63, 78, 79]. Da das Problem konvex ist, erhält man sicher das globale Optimum.

Die Optimierung der nichtlinearen Parameter ist dagegen deutlich schwieriger, da man es im Regelfall mit einem nichtkonvexen und somit multimodalen Optimierungsproblem zu tun hat. Zur Optimierung dieser Parameter eines Fuzzy-Systems werden hauptsächlich gradientenbasierte Verfahren [84, 216] sowie evolutionäre Algorithmen [59] eingesetzt. Erstere konvergieren im Allgemeinen in ein lokales Minimum, sodass gute Startwerte Voraussetzung zum Auffinden des globalen bzw. guten lokalen Optimums sind. Evolutionäre Algorithmen haben eine größere Wahrscheinlichkeit, das globale Optimum zu finden, haben aber vergleichsweise viele Einstellparameter und benötigen deutlich mehr Rechenzeit.

Darüber hinaus gibt es verschiedene Ansätze, die die gleichzeitige Optimierung beider Parametergruppen realisieren [84]. Hierzu werden zumeist die obigen Verfahren kombiniert, man führt also zunächst einige Schritte eines Gradientenverfahrens bzw. evolutionären Algorithmus durch und optimiert dann die linearen Parameter mit LS. Dies setzt man iterativ fort bis die Optimierung konvergiert.

2.2.4 Erhalt der Interpretierbarkeit

Nachdem zeitweise der Fokus auf den Approximationsfähigkeiten von Fuzzy-Systemen lag, richtet sich das Augenmerk seit Ende der neunziger Jahre wieder verstärkt auf den Erhalt der Interpretierbarkeit. Hierzu entwickelten sich Modellierungsverfahren, die die Interpretierbarkeit des Fuzzy-Systems schon im Gütekriterium berücksichtigen. Eine Übersicht über das Themenfeld geben [12, 29, 67, 86].

Eine elegante Möglichkeit, die beiden wichtigsten Kriterien Approximationsgenauigkeit und Interpretierbarkeit gleichermaßen zu berücksichtigen, ist die Definition eines Mehrzieloptimierungsproblems (MOP). Hierzu werden einige der oben beschriebenen Kriterien verwendet. Ein erster Ansatz basierte auf der Überführung des MOP in ein einzelnes Optimierungsproblem mithilfe der gewichteten Summe des quadratischen Fehlers und der Zahl an Regeln [82], berücksichtigt also vor allem die High-Level-

Interpretierbarkeit. Da das resultierende MOP sowohl die Struktur als auch die Parameter des Fuzzy-Systems optimiert, werden in letzter Zeit zumeist auf evolutionären Algorithmen basierende Mehrzieloptimierer verwendet [10, 29, 58], bei denen nicht eine einzelne Lösung sondern die Pareto-Front des MOP [120] ermittelt wird. Aus dieser Pareto-Front kann dann der Anwender die für ihn beste Lösung auswählen.

Eine hierzu verwandte Methodik sind Regularisierungsverfahren, wie sie in [145, 146] beschrieben werden. Hierbei wird das Optimierungsproblem (meist der quadratische Fehler) unter Nebenbedingungen, die die Interpretierbarkeit sicherstellen, gelöst. Die bisher verwendeten Kriterien [135, 146] beziehen sich ausschließlich auf die Zugehörigkeitsfunktionen, also auf die Low-Level-Interpretierbarkeit.

Ein weiteres Verfahren zur Erhöhung der Low-Level-Interpretierbarkeit ist das nachträgliche Zusammenfassen von Zugehörigkeitsfunktionen, das u.a. in [86, 161, 177] angewendet wird. Die Idee ist, das Fuzzy-System zunächst ohne Berücksichtigung der Interpretierbarkeit zu optimieren. Die entstehenden Zugehörigkeitsfunktionen werden anschließend auf Basis von Distanzmaßen oder durch genetische Algorithmen zusammengefasst. Oftmals arbeiten diese Verfahren iterativ, es folgt also auf eine Zusammenfassung wieder eine Optimierung der Approximationsgenauigkeit usw., bis ein Fuzzy-System mit zufriedenstellender Interpretierbarkeit vorliegt.

Insbesondere bei der Anwendung von iterativ-heuristischen Verfahren und bei Clustering, bei denen das Fuzzy-System schrittweise aufgebaut wird, werden auch statistische Informationskriterien eingesetzt [9]. Hierbei verbindet man Approximationsgenauigkeit und Interpretierbarkeit in einem gemeinsamen Kriterium. Verschlechtert sich dieses Kriterium nach einem Iterationsschritt, so wird der Aufbau des Fuzzy-Systems beendet. In [45, 211] werden hierzu verschiedene Kriterien verglichen. Den entgegengesetzten Weg gehen [68]. Zunächst wird ein Fuzzy-System mit sehr vielen Regeln initialisiert. Durch anschließendes Zusammenfassen von sehr ähnlichen Regeln auf Basis von verschiedenen Distanzmaßen wird die Regelbasis dann schrittweise verkleinert, bis die Regeln genügend stark voneinander abweichen.

In einigen Beiträgen werden auch die bereits angesprochenen Orthogonalisierungsmethoden zur Verbesserung der Interpretierbarkeit eingesetzt [40, 74, 205, 210], wobei hier ausschließlich die Reduktion der Regelbasis betrachtet wird. Die Regeln werden dabei als Regressoren eines linearen Optimierungsproblems interpretiert. Durch Orthogonalisierung dieser Regressoren schließt man dann auf die Relevanz der Regeln. Dieser Ansatz ist allerdings umstritten, wie die Ausführungen in [176] zeigen, da die Rele-

vanz der Regeln nicht immer durch die Relevanz der Regressoren bestimmt ist. Beispielsweise werden Regeln, die nur durch wenige Datenpaare angesprochen werden, frühzeitig ausgeschlossen, obwohl sie einen bestimmten, im Datensatz unterrepräsentierten Bereich des Eingangsraums abdecken.

Des Weiteren basiert die Interpretierbarkeit einer Regelbasis nicht ausschließlich auf einer möglichst geringen Anzahl von Regeln. Sind unter den verbleibenden Regeln beispielsweise solche, die auf Zugehörigkeitsfunktionen mit großer Überlappung zurückgreifen, sind zwar wenig Regeln vorhanden, die aber keine gute Unterscheidung der linguistischen Werte mehr enthalten. Entsprechend sollten also möglichst viele der Low-Level und High-Level-Kriterien, wie sie in [145, 146] beschrieben sind, erfüllt sein. Allerdings ist allen vorgestellten Verfahren gemein, dass sie sich entweder auf Low-Level- oder High-Level-Interpretierbarkeit beziehen. Verfahren, die beide Klassen von Kriterien gleichermaßen berücksichtigen, existieren dagegen bisher nicht [217].

3 Grundlagen

In diesem Kapitel sollen zunächst die Grundlagen rekurrenter Fuzzy-Systeme erläutert werden, wobei sowohl zeitkontinuierliche rekurrente Fuzzy-Systeme [4, 48] als auch zeitdiskrete rekurrente Fuzzy-Systeme [5, 97] betrachtet werden. Daran anschließend wird eine alternative Beschreibung rekurrenter Fuzzy-Systeme vorgestellt, mit der das RFS als hybrides polynomiales System beschrieben werden kann.

3.1 Zeitkontinuierliche rekurrente Fuzzy-Systeme

Zunächst sollen die Grundlagen zeitkontinuierlicher rekurrenter Fuzzy-Systeme (KRFS) vorgestellt werden, die in [4] erstmals definiert wurden. Eine ausführliche Einführung in KRFS findet sich in [48]. Die Struktur zeitkontinuierlicher rekurrenter Fuzzy-Systeme ist in Abbildung 3.1 dargestellt. Die Funktionen $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ und $\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t))$ sind jeweils vollständige Fuzzy-Systeme mit Fuzzifizierung, Inferenz und Defuzzifizierung. Somit nehmen sowohl die Eingangsvektoren $\mathbf{u}(t)$ und $\mathbf{x}(t)$ als auch die Ausgangsvektoren $\dot{\mathbf{x}}(t)$ bzw. $\mathbf{y}(t)$ reelle Werte an.

Die Dynamik des KRFS wird auf der Menge $\mathcal{Z} = \{(\mathbf{x}, \mathbf{u}) \mid x_{i, \min} \leq x_i \leq x_{i, \max}, i = 1, \dots, n, u_{p, \min} \leq u_p \leq u_{p, \max}, p = 1, \dots, m\}$ definiert und

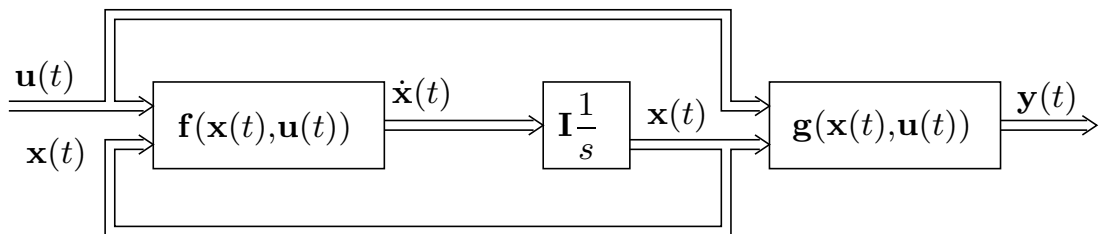


Abbildung 3.1: Blockschaltbild eines zeitkontinuierlichen rekurrenten Fuzzy-Systems.

kann linguistisch durch Regeln der Form

$$\begin{aligned} &\text{Wenn } x_1(t) = L_{j_1}^{x_1} \quad \text{und } \dots \text{ und } x_n(t) = L_{j_n}^{x_n} \\ &\text{und } u_1(t) = L_{q_1}^{u_1} \quad \text{und } \dots \text{ und } u_m(t) = L_{q_m}^{u_m}, \\ &\text{dann } \dot{x}_1(t) = L_{w_1(j,q)}^{\dot{x}_1} \text{ und } \dots \text{ und } \dot{x}_n(t) = L_{w_n(j,q)}^{\dot{x}_n} \end{aligned} \quad (3.1)$$

beschrieben werden. Hierbei sind $L_{j_i}^{x_i}$, $L_{q_p}^{u_p}$ und $L_{w_i(j,q)}^{\dot{x}_i}$ die linguistischen Werte der linguistischen Zustandsvariablen x_i , der linguistischen Eingangsvariablen u_p und der linguistischen Zustandsableitungen \dot{x}_i . Letztere beschreiben die Ableitung der Zustandsvariablen in sprachlicher Form, beispielsweise durch „positiv“, „null“ und „negativ“.

Jeder Prämisse einer Regel der Form (3.1), die durch die Index-Vektoren \mathbf{j} (Indizes der linguistischen Werte $L_{j_i}^{x_i}$ der Zustände) und \mathbf{q} (Indizes der linguistischen Werte $L_{q_p}^{u_p}$ der Eingänge) definiert ist, wird eine durch den Indexvektor \mathbf{w} (Indizes der linguistischen Werte $L_{w_i(j,q)}^{\dot{x}_i}$ der Zustandsableitungen) definierte Konklusion zugeordnet. Damit kann eine Regel der Form (3.1) auch vollständig durch die Abbildung $(\mathbf{j}, \mathbf{q}) \rightarrow \mathbf{w}(\mathbf{j}, \mathbf{q})$ beschrieben werden.

Beispiel 3.1. Als Beispiel für die Darstellung eines Systems als KRFS wird ein spezielles biologisches System, der Mutualismus, betrachtet [3, 22]. Ein Mutualismus beschreibt die symbiotische Beziehung zwischen zwei Lebewesen, bei der das eine vom anderen profitiert und umgekehrt. Ein bekanntes Beispiel ist das Zusammenleben von Landpflanzen und Pilzen. Für die Populationsentwicklungen der beiden Lebewesen können beispielhaft die folgenden Regeln aufgestellt werden:

1. Wenn die Population von Lebewesen x_1 „klein“ ist und die Population von Lebewesen x_2 „klein“ ist, dann ist die Änderung der Population des ersten Lebewesens \dot{x}_1 „positiv“ und die Änderung der Population des zweiten Lebewesens \dot{x}_2 „positiv“.
2. Wenn die Population von Lebewesen x_1 „groß“ ist und die Population von Lebewesen x_2 „klein“ ist, dann ist die Änderung der Population des ersten Lebewesens \dot{x}_1 „stark negativ“ und die Änderung der Population des zweiten Lebewesens \dot{x}_2 „leicht positiv“.
3. Wenn die Population von Lebewesen x_1 „groß“ ist und die Population von Lebewesen x_2 „groß“ ist, dann ist die Änderung der Population des ersten Lebewesens \dot{x}_1 „leicht negativ“ und die Änderung der Population des zweiten Lebewesens \dot{x}_2 „leicht negativ“.

\dot{x}_1		x_1				x_2		
		k	m	g		k	m	g
	k	lp	ln	sn	k	lp	lp	lp
	m	lp	lp	n	m	ln	lp	p
	g	lp	p	ln	g	sn	n	ln

Abbildung 3.2: Regelbasis eines KRFS, das den Mutualismus repräsentiert. Die Abkürzungen bedeuten k=„klein“, m=„mittel“, g=„groß“, l=„leicht“, s=„stark“, p=„positiv“ und n=„negativ“.

Weitere Regeln können in ähnlicher Weise angegeben werden. Werden nun die linguistischen Werte $L_{j_i}^{x_i} \in \{\text{„klein“}, \text{„mittel“}, \text{„groß“}\}$ und die linguistischen Ableitungen $L_{w_i(j)}^{\dot{x}_i} \in \{\text{„stark negativ“}, \text{„negativ“}, \text{„leicht negativ“}, \text{„leicht positiv“}, \text{„positiv“}\}$ definiert, so lässt sich Regel 2 beispielsweise durch

Wenn $x_1 = L_3^{x_1} = \text{„groß“}$ und $x_2 = L_1^{x_2} = \text{„klein“}$,
dann $\dot{x}_1 = L_1^{\dot{x}_1} = \text{„stark negativ“}$ und $\dot{x}_2 = L_4^{\dot{x}_2} = \text{„leicht positiv“}$

formalisieren. Gleichzeitig ist die Regel auch durch die Abbildung $\mathbf{j} = [3 \ 1] \rightarrow [1 \ 4] = \mathbf{w}$ vollständig erklärt. Sind alle Regeln vollständig definiert, so können sie in Form einer Regelbasis, wie in Abbildung 3.2 zu sehen, zusammengestellt werden.

Fasst man obige linguistische Werte und Ableitungen jeweils in gemeinsamen linguistischen Vektoren $\mathbf{L}_j^{\mathbf{x}}$, $\mathbf{L}_q^{\mathbf{u}}$ und $\mathbf{L}_{\mathbf{w}(j,q)}^{\dot{\mathbf{x}}}$ zusammen, so ergibt sich die folgende äquivalente Kurzform der Regeln (3.1)

$$\begin{aligned} \text{Wenn } \mathbf{x}(t) = \mathbf{L}_j^{\mathbf{x}} \quad \text{und} \quad \mathbf{u}(t) = \mathbf{L}_q^{\mathbf{u}}, \\ \text{dann } \dot{\mathbf{x}}(t) = \mathbf{L}_{\mathbf{w}(j,q)}^{\dot{\mathbf{x}}} \end{aligned} \quad (3.2)$$

in Form einer *linguistischen Differentialgleichung* mit $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$.

Um die aufgestellte Regelbasis auch quantitativ auswerten zu können, wird jedem linguistischen Wert $L_{j_i}^{x_i}$ jeder Zustandsvariablen x_i eine Zugehörigkeitsfunktion $\mu_{j_i}^{x_i}(x_i)$ auf dem Intervall $\mathcal{X}_i \subseteq \mathbb{R}$ zugeordnet. Gleiches Vorgehen führt zu den Zugehörigkeitsfunktionen $\mu_{q_p}^{u_p}(u_p)$ der Eingangsgrößen u_p . Hierbei wird die Abszisse des Maximalwerts der Zugehörigkeitsfunktion $s_{j_i}^{x_i}$ bzw. $s_{q_p}^{u_p}$ im Folgenden als *Kernposition* bezeichnet. Als Zugehörigkeitsfunktionen der linguistischen Ableitungen, also der Ausgänge von

$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$, werden Singletons $s_{w_i}^{\dot{x}_i}(\mathbf{j}, \mathbf{q})$ verwendet, die auch als *Kernpositionsableitungen* bezeichnet werden. Sie beschreiben die Ableitung des Systems in der zugehörigen Kernposition. Ähnlich wie bei den linguistischen Werten lassen sich auch die Kernpositionen und Kernpositionsableitungen in Vektorform \mathbf{s}_j^x , \mathbf{s}_q^u und $\mathbf{s}_{w(\mathbf{j}, \mathbf{q})}^{\dot{x}}$ zusammenstellen.

Zusätzlich werden an die eingangsseitigen Zugehörigkeitsfunktionen die folgenden Bedingungen gestellt [4, 5, 97]:

1. Beschränkung: $\mu_{j_i}^{x_i}(x_i) \in [0, 1]$ für alle $x_i \in \mathcal{X}_i$,
2. Konvexität:
$$\begin{cases} \mu_{j_i}^{x_i}(x_i) & \text{monoton steigend} & \forall x_i < s_{j_i}^{x_i} \\ \mu_{j_i}^{x_i}(x_i) & \text{monoton fallend} & \forall x_i > s_{j_i}^{x_i} \end{cases},$$
3. Partitionierung: $\sum_{j_i} \mu_{j_i}^{x_i}(x_i) = 1$ für alle $x_i \in \mathcal{X}_i$ und $\mu_{j_i}^{x_i}(s_{j_i}^{x_i}) = 1$ und $\mu_{j_i}^{x_i}(s_{l_i}^{x_i}) = 0$ für $j_i \neq l_i$,
4. Stetigkeit: $\mu_{j_i}^{x_i}(x_i)$ ist stetig in \mathcal{X}_i .

Außerdem wird im Rahmen der Arbeit vorausgesetzt, dass die Zugehörigkeitsfunktionen stückweise polynomial sind, was somit auch die typischerweise verwendeten Dreieck- und Rampenfunktionen [4, 5] einschließt. Zur Modellbildung werden nur diese Formen der Zugehörigkeitsfunktionen verwendet. In diesem Fall lässt sich, wie Abbildung 3.3(a) zeigt, durch obige Festlegungen das KRFS vollständig durch die Lage der Kernpositionen und Kernpositionsableitungen beschreiben. Die Kernpositionsvektoren \mathbf{s}_j^x und \mathbf{s}_q^u formen anschaulich ein Gitter im Eingangs-/Zustandsraum \mathcal{Z} . Die Kernpositionsableitungen $s_{w_i}^{\dot{x}_i}(\mathbf{j}, \mathbf{q})$ geben dann an, in welche Richtung und mit welcher Geschwindigkeit die Systemtrajektorie durch die zugehörige Kernposition führt (Abbildung 3.3(b)).

Nach diesen Festlegungen lässt sich nun auch die mathematische Form herleiten. Dies geschieht wie bereits erwähnt durch die Schritte Fuzzifizierung, Inferenz und Defuzzifizierung. Bei der Fuzzifizierung wird zunächst für jeden Wert x_i und u_p der Wahrheitsgrad zu den linguistischen Werten $L_{j_i}^{x_i}$ und $L_{q_p}^{u_p}$ bestimmt. Dieser wird beispielsweise bei einer dreieckförmigen

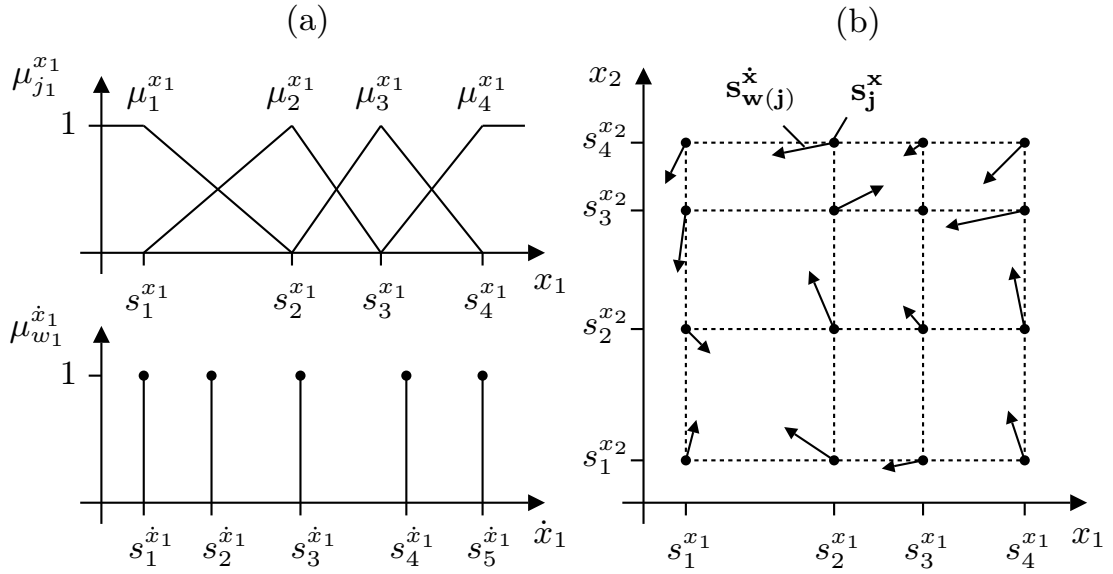


Abbildung 3.3: (a) Beispiele für Zugehörigkeitsfunktionen mit entsprechenden Kernpositionen $s_{j_1}^{x_1}$ und Kernpositionsableitungen $s_{w_1}^{\dot{x}_1}$. (b) Die Zugehörigkeitsfunktionen bilden ein Gitter im Zustands-/Eingangsraum [48].

Zugehörigkeitsfunktion über

$$\mu_{j_i}^{x_i}(x_i) = \begin{cases} \frac{(x_i - s_{j_i-1}^{x_i})}{(s_{j_i}^{x_i} - s_{j_i-1}^{x_i})}, & \text{wenn } s_{j_i-1}^{x_i} \leq x_i < s_{j_i}^{x_i}, \\ \frac{(s_{j_i+1}^{x_i} - x_i)}{(s_{j_i+1}^{x_i} - s_{j_i}^{x_i})}, & \text{wenn } s_{j_i}^{x_i} \leq x_i < s_{j_i+1}^{x_i}, \\ 0, & \text{sonst,} \end{cases} \quad (3.3)$$

bzw. die ähnlich aufgebaute Zugehörigkeitsfunktion $\mu_{q_p}^{u_p}(u_p)$ berechnet. Bei der nachfolgenden Inferenz werden zunächst die so bestimmten Wahrheitsgrade der Prämisse einer Regel aggregiert. Bei Anwendung des Produktoperators ist die Prämisse somit durch

$$\Xi_{(j,q)}(\mathbf{x}, \mathbf{u}) = \prod_{i=1}^n \mu_{j_i}^{x_i}(x_i) \prod_{p=1}^m \mu_{q_p}^{u_p}(u_p) \quad (3.4)$$

gegeben. Im Implikationsschritt wird dann auf den Wahrheitsgrad der Konklusion, also der zu der Regel gehörenden Kernpositionsableitung (Singleton), geschlossen. Schließlich werden alle Regeln akkumuliert. Als Operato-

ren für die Implikation wird das algebraische Produkt, für die Akkumulation die einfache Summe verwendet. Abschließend erfolgt die Defuzzifizierung mit der Center of Singletons Methode.

Unter Berücksichtigung der Bedingungen 1.-4. und gemäß [4, 5] ergibt sich schließlich die mathematische Beschreibung der Dynamik zu

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = \sum_{\mathbf{j}, \mathbf{q}} \mathbf{s}_{\mathbf{w}(\mathbf{j}, \mathbf{q})}^{\dot{\mathbf{x}}} \prod_{i=1}^n \mu_{j_i}^{x_i}(x_i) \prod_{p=1}^m \mu_{q_p}^{u_p}(u_p). \quad (3.5)$$

Die mathematische Beschreibung der Ausgangsfunktion $\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t))$ ist analog zur Dynamikfunktion und lautet

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) = \sum_{\mathbf{j}, \mathbf{q}} \mathbf{s}_{\mathbf{v}(\mathbf{j}, \mathbf{q})}^{\mathbf{y}} \prod_{i=1}^n \mu_{j_i}^{x_i}(x_i) \prod_{p=1}^m \mu_{q_p}^{u_p}(u_p). \quad (3.6)$$

Die Gleichungen (3.5) und (3.6) bilden ein nichtlineares Zustandsraummodell, bei dem die Vektorfelder \mathbf{f} und \mathbf{g} durch Fuzzy-Funktionen gebildet werden.

3.2 Zeitdiskrete rekurrente Fuzzy-Systeme

Zeitdiskrete rekurrente Fuzzy-Systeme (DRFS) sind zeitkontinuierlichen rekurrenten Fuzzy-Systemen strukturell sehr ähnlich, wie der Vergleich von Abbildung 3.1 und Abbildung 3.4 zeigt. Man ersetzt lediglich den Integrator durch ein Verzögerungselement, wobei aus den zeitkontinuierlichen Größen $\mathbf{x}(t)$, $\mathbf{u}(t)$, $\dot{\mathbf{x}}(t)$ und $\mathbf{y}(t)$ die zeitdiskreten Größen $\mathbf{x}(k)$, $\mathbf{u}(k)$, $\mathbf{x}(k+1)$ und $\mathbf{y}(k)$ entstehen. Die nun zeitdiskreten Fuzzy-Systeme $\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))$ und $\mathbf{g}(\mathbf{x}(k), \mathbf{u}(k))$ sind ganz analog zu den zeitkontinuierlichen aufgebaut. Es werden also wiederum linguistische Werte für die Eingänge, die Zustände im Zeitschritt k und die Zustände im Zeitschritt $k+1$ definiert. Die Dynamik wird entsprechend durch Regeln beschrieben, die nun die Form

$$\begin{aligned} &\text{Wenn } x_1(k) = L_{j_1}^{x_1} \quad \text{und } \dots \text{ und } x_n(k) = L_{j_n}^{x_n} \\ &\text{und } u_1(k) = L_{q_1}^{u_1} \quad \text{und } \dots \text{ und } u_m(k) = L_{q_m}^{u_m}, \\ &\text{dann } x_1(k+1) = L_{w_1(\mathbf{j}, \mathbf{q})}^{\dot{x}_1} \text{ und } \dots \text{ und } x_n(k+1) = L_{w_n(\mathbf{j}, \mathbf{q})}^{\dot{x}_n} \end{aligned}$$

annehmen. Auch hier lassen sich die linguistischen Werte in Vektoren zusammenfassen, was zu einer zu Gl. (3.2) äquivalenten Form führt.

Wie bereits bei zeitkontinuierlichen rekurrenten Fuzzy-Systemen eingeführt, werden den linguistischen Werten wiederum Zugehörigkeitsfunktionen zugeordnet, die die Bedingungen 1.-4. erfüllen. Im Regelfall werden wieder Dreieck- und Rampenfunktionen für die Eingänge $\mathbf{u}(k)$ und Zustände $\mathbf{x}(k)$ sowie Singletons für die Zustände $\mathbf{x}(k+1)$ im folgenden Zeitschritt verwendet. Aufgrund der Bedingungen an die Zugehörigkeitsfunktionen entstehen wiederum charakteristische Punkte, die Kernpositionen, die zu Kernpositionsvektoren zusammengefasst werden können. Der Ausgang des regelbasierten Fuzzy-Systems, dessen Zugehörigkeitsfunktionen Singletons sind, repräsentiert in diesem Fall nicht die Ableitung, sondern den Zustand $\mathbf{x}(k+1)$ im folgenden Zeitschritt. Entsprechend stellt ein Singleton keine Kernpositionsableitung dar, sondern wieder eine Kernposition, die im Folgenden zur besseren Unterscheidung auch als *Ausgangskernposition* bezeichnet wird.

Es gibt nun, abhängig von der Festlegung der Ausgangskernpositionen, zwei verschiedene Ausführungen zeitdiskreter rekurrenter Fuzzy-Systeme. Zu ihrer Unterscheidung ist die folgende Bedingung an die Ausgangszugehörigkeitsfunktionen entscheidend:

5. Rückkopplungskorrespondenz: Sei \mathcal{S}_i die Menge, die alle Kernpositionen $s_{j_i}^{x_i}$ des Zustands $x_i(k)$ enthält. Dann gilt für die Ausgangszugehörigkeitsfunktionen: $s_{w_i}^{x_i} \in \mathcal{S}_i$ für alle $w_i = 1, \dots, r_x(i)$ und $i = 1, \dots, n$, wobei $r_x(i)$ die Zahl der Regeln der i -ten Zustandsgröße bezeichnet.

Es ergeben sich damit die beiden folgenden Varianten von DRFS:

- **DRFS ohne Rückkopplungskorrespondenz:** Die Ausgangskernpositionen sind im Intervall \mathcal{X}_i ihres zugehörigen Zustands beliebig verteilt (vgl. Abbildung 3.5(a) mitte). Dies entspricht exakt der Definition der Kernpositionsableitungen, deren Werte auch beliebig sein können. Entsprechend kann man diese Version zeitdiskreter rekurrenter Fuzzy-Systeme als abgetastete zeitkontinuierliche Fuzzy-Systeme

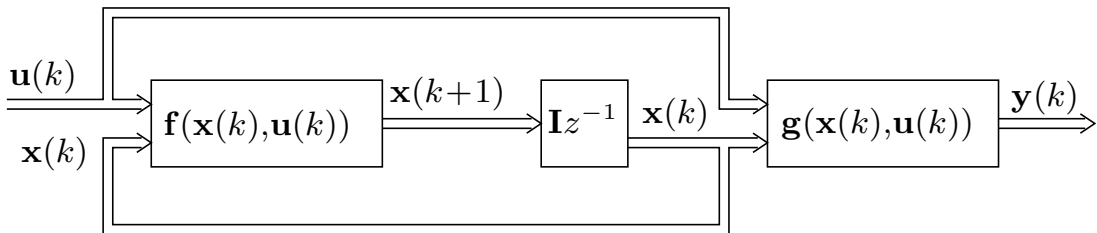


Abbildung 3.4: Blockschaltbild eines zeitdiskreten rekurrenten Fuzzy-Systems.

interpretieren. Für die quantitative Modellbildung sind diese Systeme ähnlich gut geeignet wie zeitkontinuierliche Fuzzy-Systeme. Vorteilhaft ist, dass keine Zustandsableitungen gemessen bzw. aus Messdaten der Zustände durch Ableitung gewonnen werden müssen. Nachteilig ist, dass die Interpretierbarkeit der Kernpositionsableitungen als Trajektorien durch die zugehörige Kernposition für die zeitdiskrete Version nicht mehr gilt.

- **DRFS mit Rückkopplungskorrespondenz:** Die Lage der Ausgangskernpositionen, also der Kernpositionen des Zeitschritts $k + 1$, entspricht gemäß der Rückkopplungskorrespondenz exakt der Lage der Kernpositionen der Zustände im Zeitschritt k (vgl. Abbildung 3.5(a) unten). Diese Version entspricht der ursprünglichen Definition zeitdiskreter rekurrenter Fuzzy-Systeme in [5, 97]. Diese zeitdiskrete Version eignet sich aufgrund der erheblichen Einschränkung der Entwurfsfreiheitsgrade weniger zur quantitativen als zur qualitativen Modellbildung und zur Fehlerdiagnose. Durch die Rückkopplungskorrespondenz ergibt sich eine spezielle Interpretation des DRFS [5, 97]: Operiert das DRFS ausschließlich auf den Kernpositionen, so kann es als linguistischer Zustandsautomat aufgefasst werden. Entsprechend kann für die Operation auf den Kernpositionsvektoren auch ein Zustandsgraph angegeben werden, der das Verhalten des DRFS an den Kernpositionen qualitativ beschreibt. Abbildung 3.5(b) zeigt dies für eine einfache Regelbasis. Die Kernpositionen der Zustände bilden die Zustände des Graphen. Die Regelbasis definiert dann, welche Transitionen bei welcher Kombination von Kernpositionen der Eingänge möglich sind. Zu beachten ist, dass im Gegensatz zu Zustandsautomaten [127] DRFS auch beliebige Werte zwischen den Kernpositionen annehmen können, wenn das DRFS nicht auf Kernpositionen operiert.

Die mathematische Repräsentation zeitdiskreter rekurrenter Fuzzy-Systeme ist wiederum unabhängig von der konkreten Definition der Ausgangszugehörigkeitsfunktion. Da die gleichen Operatoren für Aggregation, Implikation und Akkumulation wie bei zeitkontinuierlichen rekurrenten Fuzzy-Systemen verwendet werden, ergibt sich die Gl. (3.5) sehr ähnliche Darstellung

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) = \sum_{\mathbf{j}, \mathbf{q}} \mathbf{s}_{\mathbf{w}(\mathbf{j}, \mathbf{q})}^{\mathbf{x}} \prod_{i=1}^n \mu_{j_i}^{x_i}(x_i) \prod_{p=1}^m \mu_{q_p}^{u_p}(u_p). \quad (3.7)$$

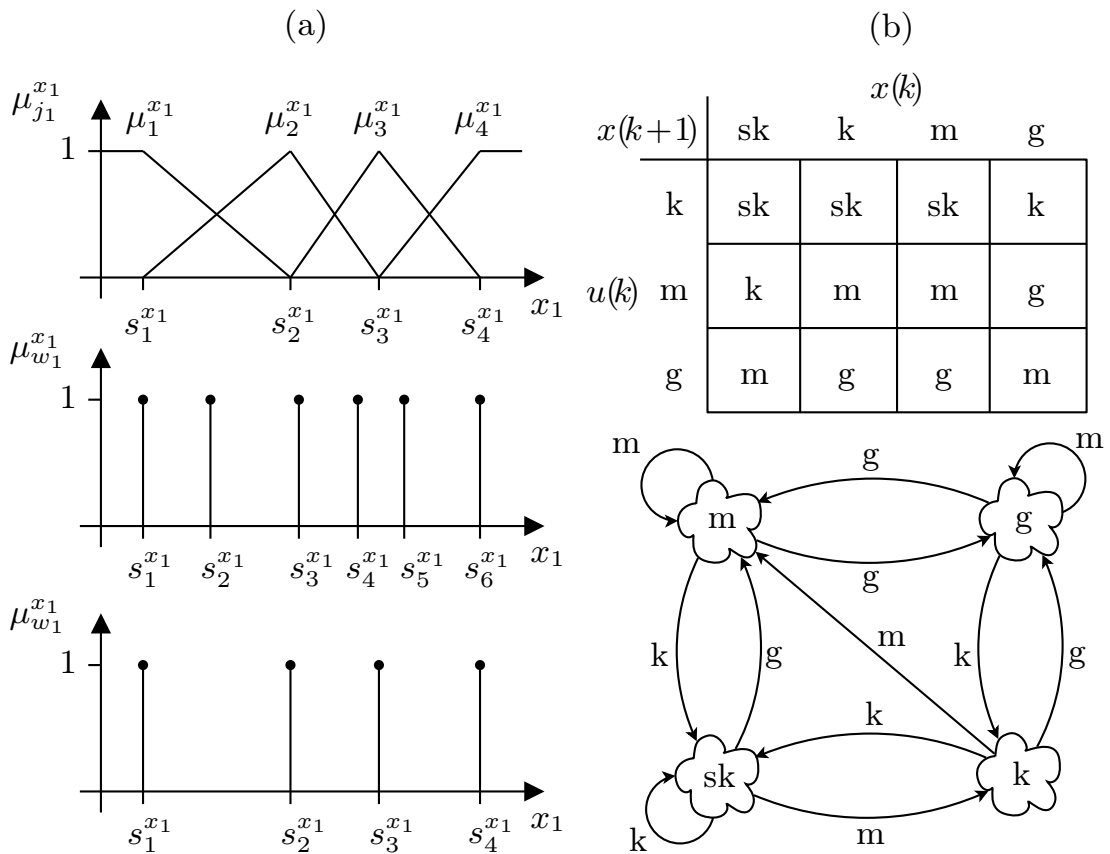


Abbildung 3.5: (a) Zugehörigkeitsfunktionen mit entsprechenden Kernpositionen s_{j1}^{x1} (oben) sowie Ausgangskernpositionen s_{w1}^{x1} für ein DRFS ohne (mitte) und mit (unten) Rückkopplungskorrespondenz. Während beim DTRFS mit Rückkopplungskorrespondenz die Kernpositionen exakt übereinstimmen, sind sie beim DTRFS ohne Rückkopplungskorrespondenz beliebig verteilt. (b) Zustandsgraph für eine Regelbasis mit einer Zustands- und Eingangsgröße und Rückkopplungskorrespondenz. Die welligen Knoten symbolisieren die unscharfen Zustände [5].

Entsprechendes gilt für die Ausgangsfunktion $\mathbf{g}(\mathbf{x}(k), \mathbf{u}(k))$.

3.3 Darstellung rekurrenter Fuzzy-Systeme als hybride Systeme

Die bis hierhin vorgestellte Darstellungsform rekurrenter Fuzzy-Systeme durch die Gleichungen (3.5) bzw. (3.7) orientiert sich an der linguistischen Interpretation des RFS und folgt den klassischen drei Schritten Fuzzifizierung, Inferenz und Defuzzifizierung.

Rekurrente Fuzzy-Systeme weisen allerdings auch eine Verbindung zu den in der Systemidentifikation vielfach eingesetzten Polynomansätzen [37, 78, 142] und zu den in den letzten Jahren verstärkt untersuchten hybriden Systemen [24, 28, 62, 89, 158] auf, die aus Gl. (3.5) nicht direkt erkennbar ist. Des Weiteren existiert ein Zusammenhang zur Theorie sequentieller Systeme nach Franke [56]. Da die Fuzzy-Logik eine Erweiterung der booleschen Logik darstellt, bestehen Querverbindungen zwischen der arithmetischen Darstellung boolescher Funktionen in Form von Shegalkin-Polynomen und der Darstellung (rekurrenter) Fuzzy-Systeme.

Im Folgenden wird eine alternative Beschreibung hergeleitet, die sowohl zeitkontinuierliche als auch zeitdiskrete RFS als hybride polynomiale Systeme darstellt. Die Teilsysteme des hybriden Systems sind, wie sich zeigen wird, affin-multilineare Funktionen, was bereits in [214, 215] für statische Fuzzy-Systeme mit Singletons als Ausgangszugehörigkeitsfunktionen und Produkt-Summe Operatoren gezeigt wurde.

Die Herleitung der hybriden Systemdarstellung RFS erfolgt in drei Schritten. Zunächst wird gezeigt, wie das RFS in Elementareinheiten, die sogenannten Hyperquader, zerlegt werden kann. Daran anschließend wird die polynomiale Darstellung eines solchen Hyperquaders hergeleitet. Schließlich wird die Darstellung des gesamten RFS angegeben.

3.3.1 Zerlegung des RFS in elementare Hyperquader

Die Idee zur Formulierung einer hybriden Systembeschreibung beruht auf einer Zerlegung des RFS. Hierzu ist der in [97] eingeführte Begriff des elementaren Hyperquaders grundlegend:

Definition 3.1. (Elementarer Hyperquader): Das Gebiet im Eingangs-/Zustandsraum $\mathcal{Z}_l = \{(\mathbf{x}, \mathbf{u}) \mid s_{j_i}^{x_i} \leq x_i \leq s_{j_i+1}^{x_i}, i = 1, \dots, n, s_{q_p}^{u_p} \leq u_p \leq s_{q_p+1}^{u_p}, p = 1, \dots, m\} \subseteq \mathbb{R}^{n+m}$ mit $j_i \in \{1, \dots, r_x(i)-1\}$, $q_p \in \{1, \dots, r_u(p)-1\}$ wird elementarer Hyperquader genannt. Der Indexvektor $\mathbf{v}_l^T = [\mathbf{j}_l^T \ \mathbf{q}_l^T]$ des l -ten elementaren Hyperquaders setzt sich aus dem jeweils kleinsten Index j_i, q_p der Kernpositionen zusammen. Hierbei bezeichnet $r_x(k)$ bzw. $r_u(k)$ die Zahl an Zugehörigkeitsfunktionen der k -ten Zustands- bzw. Eingangsvariablen.

Abbildung 3.6 zeigt eine Zerlegung des RFS aus Beispiel 3.1 in elementare Hyperquader, wobei die graue Region den Hyperquader mit Indexvektor $[1 \ 2]^T$ kennzeichnet. Neben den elementaren Hyperquadern werden zwei weitere Gebiete des Eingangs-/Zustandsraums definiert:

Definition 3.2. (Interpolationsregion): Das Gebiet im Eingangs-/Zustandsraum \mathcal{Z}_{in} , das durch die Menge $\mathcal{Z}_{in} = \{(\mathbf{x}, \mathbf{u}) \mid s_1^{x_i} \leq x_i \leq s_{r_x(i)}^{x_i}, i = 1, \dots, n, s_1^{u_p} \leq u_p \leq s_{r_u(p)}^{u_p}, p = 1, \dots, m\} \subset \mathbb{R}^{n+m}$, beschrieben wird, wird Interpolationsregion des RFS genannt.

Definition 3.3. (Extrapolationsregion): Das Gebiet im Eingangs-/Zustandsraum \mathcal{Z}_{ex} , das durch die Menge $\mathcal{Z}_{ex} = \mathcal{Z} \setminus \mathcal{Z}_{in}$ beschrieben wird, wird Extrapolationsregion des RFS genannt.

Die Interpolationsregion \mathcal{Z}_{in} beschreibt das Gebiet, in dem in jeder Dimension zwischen zwei aufeinander folgenden Zugehörigkeitsfunktionen interpoliert wird. Sie ist in Abbildung 3.6 durch die strichpunktierte Linie markiert. Außerhalb der Interpolationsregion ist die Zugehörigkeitsfunktion zumindest einer Dimension konstant eins. Dieses Gebiet ist dann die Extrapolationsregion. Mithilfe von Definition 3.1 und 3.2 lässt sich die Interpolationsregion \mathcal{Z}_{in} in

$$K_{in} = \prod_{i=1}^n (r_x(i) - 1) \cdot \prod_{p=1}^m (r_u(p) - 1)$$

elementare Hyperquader zerlegen.

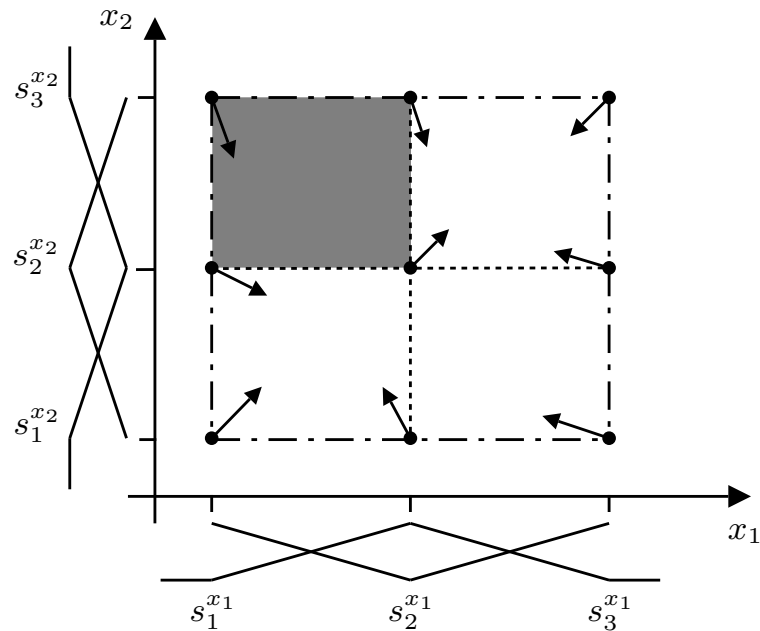


Abbildung 3.6: Beispiel für die Zerlegung des RFS aus Beispiel 1 in elementare Hyperquader. Die graue Fläche markiert den Hyperquader mit Indexvektor $[1 \ 2]^T$.

Das Gebiet des Eingangs-/Zustandsraums $\mathcal{Z}_l \subseteq \mathcal{Z}$, das den l -ten Hyperquader mit zugehörigem Indexvektor \mathbf{v}_l beschreibt, kann dabei durch

$$\mathcal{Z}_l = \{(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^{n+m} \mid \mathbf{r}_l(\mathbf{x}, \mathbf{u}) \geq \mathbf{0}\}. \quad (3.8)$$

angegeben werden, wobei

$$\mathbf{r}_l(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{x} - \mathbf{s}_{\mathbf{j}_l}^{\mathbf{x}} \\ \mathbf{u} - \mathbf{s}_{\mathbf{q}_l}^{\mathbf{u}} \\ \mathbf{s}_{\mathbf{j}_l+1}^{\mathbf{x}} - \mathbf{x} \\ \mathbf{s}_{\mathbf{q}_l+1}^{\mathbf{u}} - \mathbf{u} \end{bmatrix} \quad (3.9)$$

gilt. Auch die Extrapolationsregion kann in Hyperquader zerlegt werden. Hierzu führt man in jeder Dimension zwei virtuelle Kernpositionen $s_{-\infty}^{x_i}$ und $s_{\infty}^{x_i}$ ein, deren Wert gegen $\pm\infty$ geht. Mit diesen zusätzlichen Kernpositionen kann ein Hyperquader in der Extrapolationsregion wiederum gemäß Gl. (3.8) und (3.9) beschrieben werden.

3.3.2 Darstellung eines elementaren Hyperquaders

Nachdem im vorigen Abschnitt die Zerlegung des RFS in elementare Hyperquader angegeben wurde, soll nun die mathematische Beschreibung eines elementaren Hyperquaders hergeleitet werden. Hierfür werden zur Vereinfachung der Darstellung die Eingangsgrößen \mathbf{u} und Zustandsgrößen \mathbf{x} in einem gemeinsamen Vektor $\mathbf{z}^T := [\mathbf{x}^T \ \mathbf{u}^T]$ zusammengefasst. Außerdem wird wiederum der Operator $\delta\mathbf{x}$ verwendet, für den

$$\delta\mathbf{x} = \dot{\mathbf{x}}$$

im zeitkontinuierlichen Fall und

$$\delta\mathbf{x} = \mathbf{x}(k+1)$$

im zeitdiskreten Fall gilt.

Ein elementarer Hyperquader gemäß Definition 3.1 lässt sich mithilfe der Systemdarstellung des RFS (3.5) durch

$$\delta\mathbf{x} = \mathbf{f}_l(\mathbf{x}, \mathbf{u}) = \mathbf{f}_l(\mathbf{z}) = \sum_{\mathbf{v} \in \{\mathbf{v}_l, \mathbf{v}_l + \mathbf{1}\}} \mathbf{s}_{\mathbf{w}(\mathbf{v})}^{\delta\mathbf{z}} \prod_{i=1}^{n+m} \mu_{v_i}^{z_i}(z_i) \quad (3.10)$$

darstellen, wobei die Summe in jeder Dimension des Eingangsraums nur über die zwei Elemente \mathbf{v}_l und $\mathbf{v}_l + \mathbf{1}$ des Indexvektors gebildet wird und l den betrachteten Hyperquader kennzeichnet.

Der folgende Satz trifft nun eine Aussage über die Äquivalenz der linguistischen Darstellung (3.10) und einer polynomialen Darstellung eines elementaren Hyperquaders.

Satz 3.1. *Die folgenden Systemdarstellungen eines elementaren Hyperquaders sind äquivalent:*

1. *Linguistische Darstellung mit*

$$\delta \mathbf{x} = \mathbf{f}_l(\mathbf{x}, \mathbf{u}) = \mathbf{f}_l(\mathbf{z}) = \sum_{\mathbf{v} \in \Xi_l} \mathbf{s}_{\mathbf{w}(\mathbf{v})}^{\delta \mathbf{z}} \prod_{i=1}^{n+m} \mu_{v_i}^{z_i}(z_i). \quad (3.11)$$

2. *Polynomiale Darstellung mit*

$$\delta \mathbf{x} = \mathbf{p}_l(\mathbf{z}), \quad \mathbf{z} \in \mathcal{Z}_l, \quad (3.12)$$

mit den polynomialen Vektorfeldern

$$\mathbf{p}_l(\mathbf{z}) = \sum_{\mathbf{v} \in \Xi_l} \mathbf{s}_{\mathbf{w}(\mathbf{v})}^{\delta \mathbf{z}} \prod_{i=1}^{n+m} \mu_{v_i}^{z_i}(z_i), \quad (3.13)$$

wobei die Menge

$$\Xi_l = \{\mathbf{v} \in \mathbb{R}^{n+m} \mid \mathbf{v} = \mathbf{v}_l + \boldsymbol{\lambda}, \boldsymbol{\lambda} \in \Lambda\}, \text{ mit } \Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^{n+m} \mid \lambda_i \in \{0,1\}\}$$

die Indexvektoren aller Kernpositionen enthält, die dem l -ten Hyperquader zugeordnet sind.

Beweis: Durch Einsetzen von (3.13) in (3.12) folgt direkt die Äquivalenz von 1. und 2. Des Weiteren wird gezeigt, dass (3.12) ein polynomiales Vektorfeld ist. Aus den Annahmen und den Bedingungen 1.-4. aus Abschnitt 3.1 folgt, dass die Zugehörigkeitsfunktionen in einem elementaren Hyperquader polynomiale Funktionen sind. Da Produkte und Summen von Polynomen wiederum Polynome ergeben, sind auch die

$$p_{\mathbf{v}}(\mathbf{z}) = \prod_{i=1}^{n+m} \mu_{v_i}^{z_i}(z_i),$$

polynomiale Funktionen und somit auch die Vektorfelder eines elementaren Hyperquaders (3.13) polynomiale Funktionen. \square

Die gezeigte Äquivalenz gilt auch für Hyperquader in der Extrapolationsregion, wenn man wieder virtuelle Kernpositionen verwendet. In diesem Fall sind einige Zugehörigkeitsfunktionen konstant eins.

Beispiel 3.2. Um die hybride polynomialen Darstellung eines RFS zu erläutern, betrachten wir wiederum den Mutualismus aus Beispiel 3.1. Abbildung 3.6 zeigt die Zerlegung des Zustandsraums in vier elementare Hyperquader. Wir betrachten nun den grau markierten Hyperquader mit dem Indexvektor $\mathbf{j}_l = [1 \ 2]^T$ und der Menge

$$\Xi_l = \{[1 \ 2]^T, [2 \ 2]^T, [1 \ 3]^T, [2 \ 3]^T\}.$$

Dies führt zu

$$\begin{aligned} \mathbf{f}_l(\mathbf{x}) &= \sum_{\mathbf{j} \in \Xi_l} \mathbf{s}_{\mathbf{w}(\mathbf{j})}^{\mathbf{x}} \prod_{i=1}^n \mu_{j_i}^{x_i}(x_i) \\ &= \mathbf{s}_{\mathbf{w}([1 \ 2]^T)}^{\mathbf{x}} \mu_1^{x_1}(x_1) \mu_2^{x_2}(x_2) + \mathbf{s}_{\mathbf{w}([2 \ 2]^T)}^{\mathbf{x}} \mu_2^{x_1}(x_1) \mu_2^{x_2}(x_2) \\ &\quad + \mathbf{s}_{\mathbf{w}([1 \ 3]^T)}^{\mathbf{x}} \mu_1^{x_1}(x_1) \mu_3^{x_2}(x_2) + \mathbf{s}_{\mathbf{w}([2 \ 3]^T)}^{\mathbf{x}} \mu_2^{x_1}(x_1) \mu_3^{x_2}(x_2). \end{aligned}$$

Da wir dreieckförmige Zugehörigkeitsfunktionen verwenden, d.h. $\mu_{j_i}^{x_i}(x_i) = \frac{x_i - s_{j_i}^{x_i}}{s_{j_i+1}^{x_i} - s_{j_i}^{x_i}}$ und $\mu_{j_i+1}^{x_i}(x_i) = 1 - \mu_{j_i}^{x_i}(x_i)$ für $x_i \in \mathcal{Z}_l$, folgt

$$\mathbf{f}_l(\mathbf{x}) = \boldsymbol{\vartheta}_0 + \boldsymbol{\vartheta}_1 x_1 + \boldsymbol{\vartheta}_2 x_2 + \boldsymbol{\vartheta}_3 x_1 x_2, \quad (3.14)$$

wobei $\boldsymbol{\vartheta}_k$, $k = 0, \dots, 3$ konstante Koeffizientenvektoren sind. Gl. (3.14) stellt also ein affin-multilineares und somit polynomiales Vektorfeld dar.

Die Ergebnisse von Beispiel 3.2 lassen sich auf Systeme mit n Zuständen und m Eingängen erweitern. Wird wiederum die gängigste Form der Zugehörigkeitsfunktionen, nämlich eine dreieckförmige Zugehörigkeitsfunktion gewählt, so gilt in jedem Hyperquader eines RFS im Allgemeinen ein affin-multilineares Differential- bzw. Differenzengleichungssystem n -ter Ordnung der Form

$$\begin{aligned} \delta \mathbf{x} &= \boldsymbol{\vartheta}_0 + \sum_{i=1}^n \boldsymbol{\vartheta}_{x_i} x_i + \sum_{p=1}^m \boldsymbol{\vartheta}_{u_p} u_p + \sum_{j=2}^n \sum_{i=1}^{j-1} \boldsymbol{\vartheta}_{x_i x_j} x_i x_j \\ &\quad + \sum_{p=1}^m \sum_{i=1}^n \boldsymbol{\vartheta}_{x_i u_p} x_i + \dots + \boldsymbol{\vartheta}_{1 \dots n} x_1 \dots x_n u_1 \dots u_m = \boldsymbol{\Theta}_l \boldsymbol{\omega}. \end{aligned} \quad (3.15)$$

Hierbei gilt für die in der Matrix $\boldsymbol{\Theta}_l$ zusammengefassten Koeffizienten $\boldsymbol{\vartheta}_i$ des l -ten Hyperquaders bei entsprechender Sortierung

$$\boldsymbol{\Theta} = \mathbf{S}_{\mathbf{w}}^{\delta \mathbf{z}} \boldsymbol{\Gamma}(\mathbf{s}_{\mathbf{j}}^{\mathbf{x}}, \mathbf{s}_{\mathbf{q}}^{\mathbf{u}}). \quad (3.16)$$

Die Matrix $\mathbf{S}_{\mathbf{w}}^{\delta \mathbf{z}}$ enthält dabei alle zum Hyperquader gehörenden Kernpositionsableitungen bzw. Ausgangskernpositionen $\mathbf{s}_{\mathbf{w}(\mathbf{j}, \mathbf{q})}^{\delta \mathbf{x}}$ und $\mathbf{\Gamma}$ ist eine nicht-linear von den Kernpositionen abhängende Vektorfunktion. Die Bildung dieser Matrizen ist in Anhang A.2 angegeben. Der Vektor \mathbf{w} wiederum enthält alle multilinearen Monome der Form $x_1 \dots x_n u_1 \dots u_m$. Gl. (3.16) zeigt insbesondere, dass ein linearer Zusammenhang zwischen den Koeffizienten ϑ_i des Polynoms (3.15) und den Kernpositionsableitungen bzw. Ausgangskernpositionen $\mathbf{s}_{\mathbf{w}(\mathbf{j}, \mathbf{q})}^{\delta \mathbf{x}}$ besteht. Entsprechende Zusammenhänge lassen sich auch für Zugehörigkeitsfunktionen mit Polynomen höherer Ordnung angeben.

3.3.3 Darstellung des gesamten RFS als hybrides System

Nachdem die Zerlegung des RFS in elementare Hyperquader sowie die mathematische Darstellung eines solchen Hyperquaders eingeführt wurde, kann nun das RFS als hybrides polynomiales System dargestellt werden. Der Begriff „hybrid“ erklärt sich hierbei durch die unterschiedlichen Systembeschreibungen in Teilbereichen (Hyperquadern) des Zustandsraums. Im Allgemeinen besteht das RFS einschließlich der Extrapolationsregionen aus

$$K = \prod_{i=1}^n (r(i) + 1) \cdot \prod_{p=1}^m (r(p) + 1)$$

elementaren Hyperquadern, die jeweils für den Bereich \mathcal{Z}_l des Eingangsraums gültig sind. Entsprechend kann das RFS durch das hybride polynomiale System

$$\begin{aligned} \delta \mathbf{x} &= \mathbf{f}_l(\mathbf{z}), & \mathbf{z} &\in \mathcal{Z}_l, \\ &= \mathbf{f}_l(\mathbf{x}, \mathbf{u}), & [\mathbf{x}^T \ \mathbf{u}^T]^T &\in \mathcal{X}_l \times \mathcal{U}_l, \quad l = 1, \dots, K \end{aligned} \quad (3.17)$$

beschrieben werden.

3.3.4 Diskussion der Darstellung

Abschließend soll die vorgestellte hybride Systemdarstellung eines RFS bezüglich der bisher üblichen linguistisch basierten Beschreibung eingeordnet werden. Prinzipiell sind beide Darstellungen äquivalent und können gleichermaßen zur mathematischen Beschreibung RFS verwendet werden. Der Vorteil der linguistisch basierten Darstellung liegt in ihrer einfachen Interpretierbarkeit, da hiermit die das System beschreibenden Regeln direkt in

eine mathematische Darstellung umgesetzt werden. Allerdings zeigt sich, dass die hybride Systemdarstellung in einigen Fällen Vorteile bietet, die im Folgenden kurz diskutiert und in den folgenden Kapiteln an einigen Stellen vertieft werden:

- *Stabilitätsanalyse in RFS*: Die Darstellung von RFS als hybride polynomiale Systeme ermöglicht die Anwendung bekannter Verfahren zur Stabilitätsanalyse solcher Systeme. Dies wird in Kapitel 4 weiter beleuchtet.
- *Einbringen von Vorwissen in die Modellbildung*: Durch die hybride Darstellung mit multilinearen Vektorfeldern eröffnet sich die Möglichkeit, Vorwissen über die Systemstruktur sowie über das Systemverhalten in bestimmten Bereichen des Zustandsraums in die Modellbildung zu integrieren. Des Weiteren können mithilfe der Darstellung datengestützt Systemeigenschaften wie relevante Eingangs- und Zustandsgrößen extrahiert werden. Hierauf wird in Kapitel 5 näher eingegangen.

4 Stabilitätsanalyse rekurrenter Fuzzy-Systeme

Eine der wichtigsten Problemstellungen bei der Untersuchung von dynamischen nichtlinearen Systemen ist die Frage nach der Stabilität ihrer Ruhelagen. Dies gilt auch für rekurrente Fuzzy-Systeme. Entsprechend existieren sowohl für zeitdiskrete als auch für zeitkontinuierliche RFS bereits Ergebnisse zur Existenz von Ruhelagen und deren Stabilitätseigenschaften [4, 48, 97, 98].

Für zeitdiskrete RFS mit Rückkopplungskorrespondenz wurde die Existenz und Stabilität von Ruhelagen in [97, 98] untersucht. Die Stabilitätsbedingungen sind hierbei eng mit der Regelbasis verbunden, d.h., die Stabilität wurde auf der linguistischen Ebene des Fuzzy-Systems untersucht. Allerdings ist die Gültigkeit der Bedingungen auf eine eingeschränkte Klasse DRFS mit Rückkopplungskorrespondenz, die regelstetigen standardisierten DRFS [97], beschränkt.

Auch zeitkontinuierliche RFS wurden bereits bezüglich Ruhelagen und deren Stabilität untersucht, wobei zwischen Ruhelagen auf Kernpositionen und Ruhelagen innerhalb elementarer Hyperquader unterschieden wurde [4, 48]. Für Ruhelagen auf Kernpositionen wurden hinreichende Bedingungen für ihre asymptotische Stabilität angegeben, die wiederum direkt aus der Regelbasis ablesbar sind. Des Weiteren wurden Kriterien angegeben, mit denen die Existenz bzw. Nichtexistenz von Ruhelagen innerhalb elementarer Hyperquader nachgewiesen werden kann. Es wurde aber keine Aussage über deren Stabilität getroffen. In [183] wurden Stabilitätsbedingungen für einen rekurrenten Fuzzy-Systemen sehr ähnlichen Typ von Fuzzy-Systemen vorgestellt, die sowohl im Zeitdiskreten als auch im Zeitkontinuierlichen formuliert werden können. Die Bedingungen sind eng an die Eckpunkte und Kanten lokaler Gebiete des Fuzzy-Systems geknüpft. Allerdings wird die Zahl an Stabilitätsbedingungen schon bei niedrigen Systemdimensionen sehr groß.

Im Folgenden wird ein neuer Ansatz zur Stabilitätsanalyse vorgeschlagen [173, 174], der die Analyse von zeitkontinuierlichen und zeitdiskreten rekurrenten Fuzzy-Systemen in einem gemeinsamen Rahmen erlaubt. Hierzu

wird die im vorigen Abschnitt vorgestellte Darstellung rekurrenter Fuzzy-Systeme als hybrides polynomiales System verwendet. Diese ermöglicht die Anwendung von Methoden zur Analyse hybrider polynomialer Systeme, wie der Methode der Sum-of-Squares (SOS) [148, 149]. Der Ansatz bietet die Möglichkeiten auch Ruhelagen in elementaren Hyperquadranten zu untersuchen. Da die analytische Berechnung der Ruhelagen in RFS insbesondere bei Systemen höherer Ordnung nicht mehr einfach möglich ist, kann mithilfe eines zweischrittigen Verfahrens auch die Stabilität von Ruhelagen nachgewiesen werden, die *zuvor unbekannt* sind, d.h., deren Zahlenwert nicht bekannt ist.

4.1 Grundlagen

Zunächst werden kurz einige Grundlagen, wie die Stabilität hybrider Systeme und die Sum-of-Squares Methodik, die in den folgenden Abschnitten benötigt werden, erläutert.

4.1.1 Stabilität hybrider Systeme

Dieser Abschnitt stellt Kriterien zur Stabilitätsanalyse hybrider Systeme vor. Das bekannte Stabilitätskriterium nach Lyapunov [101] lautet in einer generalisierten Form für zeitkontinuierliche und zeitdiskrete Systeme:

Satz 4.1. *Sei \mathbf{x}_e eine Ruhelage des Systems $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ (bzw. $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k))$) und $\mathcal{D} \subseteq \mathbb{R}^n$ ein Gebiet des Zustandsraums, das \mathbf{x}_e enthält. Sei $V : \mathcal{D} \rightarrow \mathbb{R}$ eine stetig differenzierbare Funktion, sodass*

$$V(\mathbf{x}_e) < V(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{D} \setminus \{\mathbf{x}_e\},$$

$$\dot{V}(\mathbf{x}) \leq 0 \quad (\Delta V(\mathbf{x}) = V(\mathbf{x}(k+1)) - V(\mathbf{x}(k)) \leq 0), \quad \forall \mathbf{x} \in \mathcal{D}.$$

Dann ist die Ruhelage \mathbf{x}_e stabil im Sinne von Lyapunov. Falls

$$\dot{V}(\mathbf{x}) < 0 \quad (\Delta V(\mathbf{x}) < 0), \quad \forall \mathbf{x} \in \mathcal{D} \setminus \{\mathbf{x}_e\},$$

gilt, ist die Ruhelage \mathbf{x}_e asymptotisch stabil.

Beweis: Der Beweis findet sich z.B. in [101]. □

Satz 4.1 setzt voraus, dass die Ruhelage \mathbf{x}_e des untersuchten Systems bekannt ist. Zunächst wird davon ausgegangen, dass diese Annahme gilt. Somit kann

$$\mathbf{0} = \mathbf{f}_p(\mathbf{x}_e) \quad (\mathbf{x}_e = \mathbf{f}_p(\mathbf{x}_e)), \quad \mathbf{x}_e \in \mathcal{X}_p,$$

nach \mathbf{x}_e aufgelöst werden und \mathbf{x}_e ist eine Ruhelage von (3.17), die im p -ten elementaren Hyperquader liegt. Wendet man nun Satz 4.1 auf das System (3.17) an, so ergeben sich die folgenden Bedingungen für die Stabilität von \mathbf{x}_e :

Satz 4.2. *Sei \mathbf{x}_e eine Ruhelage des hybriden Systems $\dot{\mathbf{x}} = \mathbf{f}_l(\mathbf{x})$ (bzw. $\mathbf{x}(k+1) = \mathbf{f}_l(\mathbf{x}(k))$), $\mathbf{x} \in \mathcal{X}_l$, $l = 1, \dots, K$, und $\mathcal{D} \subseteq \mathbb{R}^n$ ein Gebiet, das \mathbf{x}_e enthält. Sei $V : \mathcal{D} \rightarrow \mathbb{R}$ eine stetig differenzierbare Funktion, sodass*

$$V(\mathbf{x}) > V(\mathbf{x}_e), \quad \mathbf{x} \in \bigcup_{l=1}^K \mathcal{X}_l \setminus \{\mathbf{x}_e\}, \quad (4.1)$$

$$\frac{\partial V}{\partial \mathbf{x}} \mathbf{f}_l(\mathbf{x}) < 0 \quad (\text{bzw. } \Delta V(\mathbf{x}) < 0), \quad \mathbf{x} \in \mathcal{X}_l \setminus \{\mathbf{x}_e\}, \quad l = 1, \dots, K, \quad (4.2)$$

gilt, dann ist die Ruhelage \mathbf{x}_e asymptotisch stabil.

Beweis: Satz 4.2 folgt direkt aus Satz 4.1. □

Betrachtet man nun als Spezialfall hybride Systeme mit polynomialen Vektorfeldern $\mathbf{f}_l(\mathbf{x})$, wie sie nach Abschnitt 3.3.3 auch RFS darstellen, und wählt polynomiale Lyapunov-Funktionen $V(\mathbf{x})$, dann werden die Bedingungen (4.1)-(4.2) offensichtlich zu polynomialen Ungleichungen. Dies wiederum ermöglicht die Anwendung der SOS-Zerlegung, womit die Suche nach einer polynomialen Funktion $V(\mathbf{x})$ als konvexes Optimierungsproblem dargestellt werden kann. Hierauf soll im folgenden Abschnitt eingegangen werden.

4.1.2 SOS-Methodik

Im Folgenden soll eine kurze Einführung in die SOS-Methodik gegeben werden. Eine ausführliche Darstellung findet sich z.B. in [148, 149]. Eine erste Anwendung der SOS-Methodik auf Fuzzy-Systeme wurde in [190] zur Stabilitätsanalyse von polynomialen Fuzzy-Systemen vorgestellt. Wir beginnen mit der folgenden Definition:

Definition 4.1 (Polynom). Ein Polynom $p(\mathbf{x})$ in $\mathbf{x} = [x_1, \dots, x_n]^T$ der Ordnung δ^* und Koeffizienten $c_{\alpha_1, \dots, \alpha_n}$ ist eine Linearkombination von Monomen $x_1^{\alpha_1} \cdot \dots \cdot x_n^{\alpha_n}$ der Form

$$p(\mathbf{x}) = \sum_{\alpha_1 + \dots + \alpha_n \leq \delta^*} c_{\alpha_1, \dots, \alpha_n} \cdot x_1^{\alpha_1} \cdot \dots \cdot x_n^{\alpha_n}.$$

Die Menge der Polynome in den Komponenten von \mathbf{x} wird mit $\mathbb{R}[\mathbf{x}]$ bezeichnet.

Definition 4.2 (SOS-Polynom). Die Menge der SOS-Polynome $\Sigma[\mathbf{x}]$ ist die Menge aller Polynome $p(\mathbf{x})$, die äquivalent durch eine endliche Summe von quadrierten Polynomen $p_i(\mathbf{x})$ dargestellt werden kann, d.h.,

$$\Sigma[\mathbf{x}] := \left\{ p(\mathbf{x}) \in \mathbb{R}[\mathbf{x}] \mid p(\mathbf{x}) = \sum_{i=1}^N p_i^2(\mathbf{x}), p_i(\mathbf{x}) \in \mathbb{R}[\mathbf{x}] \right\}.$$

Da ein SOS-Polynom durch die Summe von quadrierten Polynomen beschrieben werden kann, ist es offenbar nichtnegativ für alle $\mathbf{x} \in \mathbb{R}^n$. Daraus folgt, dass die Forderung nach globaler Nichtnegativität eines Polynoms ersetzt werden kann durch die Forderung, dass das betrachtete Polynom zur Menge der SOS-Polynome $\Sigma[\mathbf{x}]$ gehört. Es gilt also:

$$p(\mathbf{x}) \in \Sigma[\mathbf{x}] \Rightarrow p(\mathbf{x}) \geq 0, \forall \mathbf{x} \in \mathbb{R}^n.$$

Der Umkehrschluss gilt dagegen im Allgemeinen nicht [149]. Vielmehr handelt es sich bei der Forderung nach Zugehörigkeit eines Polynoms zur Menge der SOS-Polynome um eine Relaxierung der Forderung, dass ein Polynom nichtnegativ im Definitionsbereich ist. Die Überprüfung, ob ein Polynom für alle $\mathbf{x} \in \mathbb{R}^n$ zur Menge der SOS-Polynome gehört, stellt ein konvexes Validierungsproblem dar und kann sehr effizient mithilfe semidefiniter Programmierung gelöst werden [149].

Mit der obigen Methodik kann die Nichtnegativität eines Polynoms nachgewiesen werden. Allerdings will man in manchen Fällen nicht nur Nichtnegativität nachweisen, sondern zeigen, dass das Polynom $p(\mathbf{x})$ positiv definit ist, d.h., es soll gelten $p(\mathbf{0}) = 0$ und $p(\mathbf{x}) > 0$ für alle $\mathbf{x} \neq \mathbf{0}$. Um auch dieses Problem in ein SOS-Problem umzuwandeln, definiert man ein positiv definites Hilfspolynom wie folgt:

$$\varepsilon(\mathbf{x}) = \varepsilon_1 x_1^{2d} + \dots + \varepsilon_n x_n^{2d}, \quad \varepsilon_i > 0. \quad (4.3)$$

Falls nun eine Lösung des SOS-Problems

$$p(\mathbf{x}) - \varepsilon(\mathbf{x}) \in \Sigma[\mathbf{x}] \quad (4.4)$$

existiert, folgt daraus, dass $p(\mathbf{x}) \geq \varepsilon(\mathbf{x})$ ist. Da hiermit $p(\mathbf{x})$ größer oder gleich einem positiv definitem Polynom ist, ist dann auch $p(\mathbf{x})$ positiv definit ist. Abbildung 4.1(a) illustriert dies.

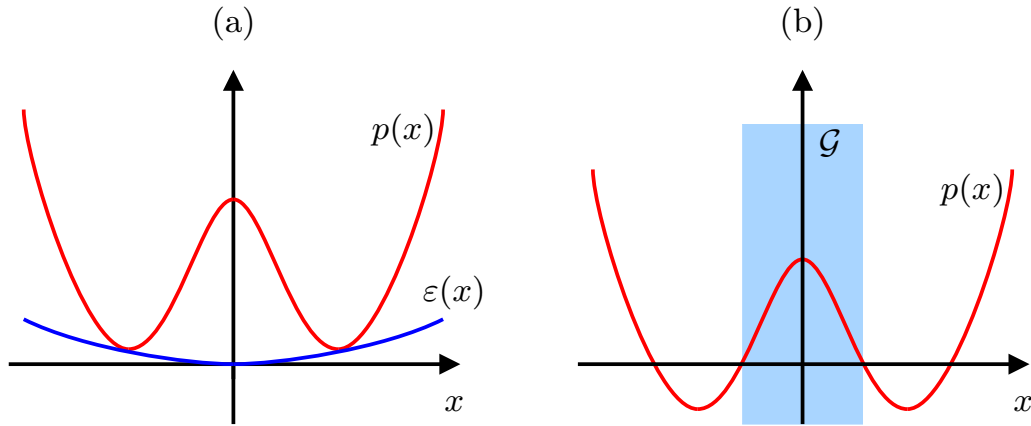


Abbildung 4.1: (a) Polynom $p(x)$ und positiv definites Hilfspolynom $\varepsilon(x)$. Da $p(x) \geq \varepsilon(x)$ gilt, ist auch $p(x)$ positiv definit. (b) Polynom $p(x)$ und Gebiet \mathcal{G} , in dem das Polynom positiv ist.

Das Auffinden einer SOS-Zerlegung ist hinreichend für die Nichtnegativität eines Polynoms im gesamten \mathbb{R}^n . Insbesondere bei der Formulierung der Stabilitätsbedingungen hybrider Systeme in Satz 4.2 treten aber Bedingungen auf, die nur die lokale Nichtnegativität eines Polynoms $p(\mathbf{x})$ in einem gegebenen Gebiet \mathcal{G} fordern. Einen solchen Fall zeigt Abbildung 4.1(b). Um nun lokale Nichtnegativität eines Polynoms nachzuweisen, kann der folgende Satz verwendet werden, der auf einer Verallgemeinerung der S-Prozedur basiert [148, 149]:

Satz 4.3. *Gegeben sei ein Polynom $f(\mathbf{x})$ und N Polynome p_1, p_2, \dots, p_N . Falls N Polynome $a_1, a_2, \dots, a_N \in \Sigma[\mathbf{x}]$ existieren, sodass*

$$f(\mathbf{x}) - \sum_{k=1}^N a_k(\mathbf{x})p_k(\mathbf{x}) \in \Sigma[\mathbf{x}]$$

gilt, dann gilt

$$\bigcap_{k=1}^N \{\mathbf{x} \mid p_k(\mathbf{x}) \geq 0\} \subseteq \{\mathbf{x} \mid f(\mathbf{x}) \geq 0\}.$$

Beweis: Der Beweis findet sich in [85]. □

Satz 4.3 wird in zwei speziellen Ausführungen zur Stabilitätsanalyse in RFS verwendet. Die erste Ausführungsform betrifft Gebiete, die durch Ungleichungsnebenbedingungen, die zweite betrifft Gebiete, die durch Gleichungsnebenbedingungen beschrieben werden. Erstens erlaubt Satz 4.3 die Überprüfung, ob

$$f(\mathbf{x}) \geq 0, \quad \forall \mathbf{x} \in \mathcal{G}_1, \quad (4.5)$$

mit

$$\mathcal{G}_1 = \{\mathbf{x} \mid p_k(\mathbf{x}) \geq 0, k = 1, \dots, N\}.$$

Dies geschieht durch Relaxierung der Forderung nach Nichtnegativität (4.5) durch

$$\begin{aligned} f(\mathbf{x}) - \sum_{k=1}^N a_k(\mathbf{x})p_k(\mathbf{x}) &\in \Sigma[\mathbf{x}], \\ a_k(\mathbf{x}) &\in \Sigma[\mathbf{x}], \quad k = 1, \dots, N. \end{aligned}$$

Die Funktionen $a_k(\mathbf{x})$ sind polynomiale Hilfsfunktionen, deren Ordnung vorgegeben wird. Ihre Koeffizienten sind freie Parameter des Validierungsproblems.

Zweitens erlaubt Satz 4.3 die Überprüfung der Bedingungen in Gebieten

$$\mathcal{G}_2 = \{\mathbf{x} \mid q_k(\mathbf{x}) = 0, k = 1, \dots, M\},$$

wie in [202] gezeigt wird. Dann sichert die SOS-Bedingung

$$f(\mathbf{x}) - \sum_{k=1}^M b_k(\mathbf{x})q_k(\mathbf{x}) \in \Sigma[\mathbf{x}],$$

dass

$$f(\mathbf{x}) \geq 0, \quad \forall \mathbf{x} \in \mathcal{G}_2$$

gilt. Auch hier sind die Funktionen $b_k(\mathbf{x})$ polynomiale Hilfsfunktionen mit fester Ordnung, wobei deren Koeffizienten freie Parameter sind.

4.2 Stabilität bekannter Ruhelagen

Nachdem die Grundlagen bekannt sind, wird nun die Stabilität von Ruhelagen in rekurrenten Fuzzy-Systemen untersucht. Zunächst wird der einfachere Fall, dass die Ruhelage bekannt ist, betrachtet. Ohne Einschränkung der Allgemeinheit wird $\mathbf{x}_e = \mathbf{0}$ vorausgesetzt, was durch eine entsprechende Transformation immer erreichbar ist. Wie in [148] für allgemeine hybride polynomiale Systeme mit bekannter Ruhelage gezeigt wurde, können die Stabilitätsbedingungen (4.1)-(4.2) mithilfe der SOS-Methodik relaxiert werden. Um $V(\mathbf{0}) = 0$ zu garantieren, wird

$$V(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$$

mit fester Ordnung 2δ als polynomiale Lyapunov-Funktion angesetzt. Unter Verwendung von (4.3)-(4.4) und Satz 4.3 erhält man

Satz 4.4. *Sei $\mathbf{x}_e = \mathbf{0}$ eine Ruhelage des hybriden Systems $\dot{\mathbf{x}} = \mathbf{f}_l(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}_l$, $l = 1, \dots, K$. Falls eine polynomiale Funktion V und Hilfspolynome $a_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}]$, $l = 1, \dots, K$; $k = 1, \dots, 2n$, existieren, sodass*

$$V(\mathbf{x}) - \varepsilon(\mathbf{x}) \in \Sigma[\mathbf{x}], \quad (4.6)$$

$$-\frac{\partial V}{\partial \mathbf{x}} \mathbf{f}_l(\mathbf{x}) - \sum_{k=1}^{2n} a_{lk}(\mathbf{x}) r_{lk}(\mathbf{x}) - \varepsilon(\mathbf{x}) \in \Sigma[\mathbf{x}], \quad l = 1, \dots, K, \quad (4.7)$$

wobei $r_{lk}(\mathbf{x})$ die k -te Komponente von \mathbf{r}_l aus (3.9) bezeichnet und $\varepsilon(\mathbf{x})$ gemäß Gl. (4.3) definiert ist, dann ist die Ruhelage $\mathbf{x}_e = \mathbf{0}$ asymptotisch stabil.

Beweis: Es ist zu zeigen, dass die SOS-Bedingungen (4.6)-(4.7) hinreichend für die Bedingungen (4.1)-(4.2) aus Satz 4.2 sind. Die SOS-Bedingung (4.6) ist hinreichend für Bedingung (4.1). Da gefordert wird, dass $V(\mathbf{x})$ positiv definit ist, wird, wie bereits in Gl. (4.3) gezeigt, ein geeignetes Hilfspolynom $\varepsilon(\mathbf{x})$ definiert. Damit garantiert Bedingung (4.6), dass $V(\mathbf{x}) > 0$ für alle \mathbf{x} mit Ausnahme der Ruhelage $\mathbf{x}_e = \mathbf{0}$ ist. Die Erfüllung von Bedingung (4.2), die fordert, dass die Lyapunov-Funktion $V(\mathbf{x})$ in jedem elementaren Hyperquader abnimmt, wird durch die SOS-Bedingung (4.7) garantiert. Hierbei wird Satz 4.3 verwendet, um die Bedingung auf den zugehörigen elementaren Hyperquader einzuschränken und $\varepsilon(\mathbf{x})$ ist wiederum ein Hilfspolynom, um strikte Negativität zu gewährleisten. \square

Analog zum zeitkontinuierlichen Fall kann Satz 4.4 auch im zeitdiskreten Fall verwendet werden, wenn in Bedingung (4.7) der Ausdruck $-\frac{\partial V}{\partial \mathbf{x}} \mathbf{f}_l(\mathbf{x})$ durch $V(\mathbf{x}) - V(\mathbf{f}_l(\mathbf{x}))$ ersetzt wird.

Die SOS-Beschränkungen (4.6)-(4.7) beschreiben eine Untermenge von gültigen polynomialen Lyapunov-Funktionen für das System (3.17). Falls diese Menge nicht leer ist, ist \mathbf{x}_e asymptotisch stabil. Falls die Menge leer ist, ist keine Aussage bezüglich der Stabilität von \mathbf{x}_e möglich.

4.3 Stabilität unbekannter Ruhelagen

Wie bereits erwähnt, ist die Annahme, dass die Ruhelage eines RFS exakt bekannt ist, oftmals nicht gerechtfertigt. Um auch in einem solchen Fall die Stabilität untersuchen zu können, gibt es prinzipiell zwei verschiedene

Möglichkeiten. Die erste besteht darin, zunächst die Ruhelagen in jedem Hyperquader numerisch zu bestimmen und die daraus erhaltenen Ruhelagen mit Satz 4.4 zu untersuchen. Tatsächlich lassen sich die Ruhelagen bei Systemen niedriger Ordnung auch noch relativ einfach bestimmen. Insbesondere bei RFS höherer Dimensionen ($n > 3$) können die Ruhelagen aber nur noch mit hohem Aufwand bestimmt werden [119]. Dies ist insbesondere der Tatsache geschuldet, dass numerische Verfahren zur Nullstellensuche Startwerte benötigen, um die Ruhelage zu berechnen. Da das RFS aber in jedem Hyperquader unterschiedliche Systembeschreibungen enthält, kann sogar ein guter Startwert zur Konvergenz in eine Ruhelage außerhalb des Hyperquaders führen, die dann keine Ruhelage des RFS ist. Dann muss die Nullstellensuche mit einem neuen Startwert fortgesetzt werden usw. Es ist deshalb wünschenswert, Aussagen über die Stabilität einer Ruhelage zu treffen, ohne ihre Lage genau zu kennen.

Hierzu wird im Folgenden ein zweischrittiges Verfahren vorgeschlagen, das eine Stabilitätsaussage für unbekannte Ruhelagen von RFS erlaubt. Im ersten Schritt wird ein Polynom $V(\mathbf{x})$ konstruiert, das zwar die notwendigen Bedingungen für eine Lyapunov-Funktion $V(\mathbf{x})$ für das RFS erfüllt. Allerdings sind die hinreichenden Bedingungen nicht erfüllt. Im zweiten Schritt wird $V(\mathbf{x})$ dann hinsichtlich hinreichender Bedingungen überprüft, sodass asymptotische Stabilität garantiert ist.

Im Folgenden wird der Ansatz zur Stabilitätsanalyse zunächst für den zeitkontinuierlichen Fall erklärt. Für den zeitdiskreten Fall ergeben sich analoge Ergebnisse, die anschließend diskutiert werden.

Schritt 1: Bestimmung möglicher Lyapunov-Funktionen.

Unter Verwendung der SOS-Methodik sucht man nach Polynomen $V(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ und $a_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}]$, sodass

$$V(\mathbf{x}) \in \Sigma[\mathbf{x}], \quad (4.8)$$

$$-\frac{\partial V}{\partial \mathbf{x}} \mathbf{f}_l(\mathbf{x}) - \sum_{k=1}^{2n} a_{lk}(\mathbf{x}) r_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}], \quad l = 1, \dots, K \quad (4.9)$$

gilt. Die Bedingungen (4.8)-(4.9) implizieren somit

- $V(\mathbf{x}) \geq 0$ für $\mathbf{x} \in \mathbb{R}^n$,
- $\dot{V}(\mathbf{x}) \leq 0$ für $\mathbf{x} \in \mathcal{X}$,

sodass $V(\mathbf{x})$ eine mögliche polynomiale Lyapunov-Funktion ist. Falls die Menge der Polynome $V(\mathbf{x})$, $a_{lk}(\mathbf{x})$ leer ist, kann ähnlich wie in Abschnitt 4.2 keine Aussage zur Stabilität getroffen werden.

Der Fall, dass die Menge von möglichen polynomialen Lyapunov-Funktionen nicht leer ist, ist notwendig, aber nicht hinreichend dafür, dass $V(\mathbf{x})$ eine Lyapunov-Funktion ist, da hierfür gemäß Satz 4.2 $V(\mathbf{x}) > V(\mathbf{x}_e)$ für alle $\mathbf{x} \neq \mathbf{x}_e$ gelten muss. Dies ist kein Problem, wenn die Ruhelage bekannt ist. In diesem Fall kann die Ruhelage in den Ursprung transformiert und ein entsprechendes Hilfspolynom $\varepsilon(\mathbf{x})$ definiert werden, sodass die strikte Positivität von $V(\mathbf{x})$ sichergestellt ist. Da aber nun davon ausgegangen wird, dass die Ruhelage unbekannt ist, kann sie weder in den Ursprung transformiert werden noch kann ein Hilfspolynom definiert werden. Stattdessen wird ein zusätzlicher Schritt eingeführt, mit dem $V(\mathbf{x}) > V(\mathbf{x}_e)$ verifiziert wird. Soll des Weiteren asymptotische Stabilität nachgewiesen werden, ist nach Satz 4.2 auch strikte Negativität von $\dot{V}(\mathbf{x})$ zu fordern. Wie zuvor kann auch hier kein Hilfspolynom definiert werden. Stattdessen wird ein zweiter Verifikationsschritt verwendet, um $\dot{V}(\mathbf{x}) < 0$ sicherzustellen. Sind beide Verifikationsschritte für ein ausgewähltes $V^*(\mathbf{x})$, das zur Menge der möglichen $V(\mathbf{x})$ gemäß den Bedingungen (4.8)-(4.9) gehört, erfolgreich, so ist asymptotische Stabilität bewiesen.

Schritt 2: Verifikation einer möglichen Lyapunov-Funktion.

Im Folgenden soll gezeigt werden, dass für die mögliche Lyapunov-Funktion $V(\mathbf{x}_e) < V(\mathbf{x})$ sowie $\dot{V}(\mathbf{x}) < 0$ für $\mathbf{x} \neq \mathbf{x}_e$ erfüllt ist. Falls dies gelingt, ist das in Schritt 1 ermittelte $V(\mathbf{x})$ tatsächlich eine Lyapunov-Funktion. Zunächst wird die Verifikation von $\dot{V}(\mathbf{x}) < 0$ betrachtet.

Schritt 2.1: Verifikation von $\dot{V}(\mathbf{x}) < 0$.

Um die asymptotische Stabilität zu garantieren, muss $\dot{V}(\mathbf{x}) < 0$ für alle $\mathbf{x} \neq \mathbf{x}_e$ gelten. Hierfür wird der Fall, dass $\dot{V}(\mathbf{x}) = 0$ ist, wenn $\mathbf{f}_l(\mathbf{x}) \neq \mathbf{0}$, $l = 1, \dots, K$ ist, ausgeschlossen. Dann tritt $\dot{V}(\mathbf{x}) = 0$ tatsächlich nur in den Ruhelagen auf. Abbildung 4.2 illustriert einen solchen auszuschließenden Fall, bei dem neben der Ruhelage \mathbf{x}_e ein weiterer Punkt existiert, indem $\dot{V}(\mathbf{x}) = 0$ gilt. Durch Lösen der folgenden SOS-Optimierungsprobleme für $l = 1, \dots, K$ und $i = 1, \dots, n$ kann ein solcher Fall überprüft werden:

$$\max_{f_{\min,l,i}} \text{, sodass} \\ -f_{\min,l,i} + f_{l,i}(\mathbf{x}) - \frac{\partial V}{\partial \mathbf{x}} \mathbf{f}_l(\mathbf{x}) c_{li1}(\mathbf{x}) - \sum_{k=1}^{2n} a_{lk}(\mathbf{x}) r_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}] \quad (4.10)$$

und

$\min f_{\max,l,i}$, sodass

$$f_{\max,l,i} - f_{l,i}(\mathbf{x}) - \frac{\partial V}{\partial \mathbf{x}} \mathbf{f}_l(\mathbf{x}) c_{li2}(\mathbf{x}) - \sum_{k=1}^{2n} a_{lk}(\mathbf{x}) r_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}], \quad (4.11)$$

mit $c_{li1}(\mathbf{x}), c_{li2}(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$, $a_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}]$. Die obigen SOS-Probleme ermitteln eine untere Grenze $f_{\min,l,i}$ und eine obere Grenze $f_{\max,l,i}$ für die Komponenten $f_{l,i}(\mathbf{x})$ des Vektorfelds $\mathbf{f}_l(\mathbf{x})$ in den Punkten, in denen $\dot{V}(\mathbf{x}) = 0$ ist. Ergibt sich nun

$$f_{\max,l,i} = f_{\min,l,i} = 0 \quad (4.12)$$

für alle l, i , so sind alle Punkte $\mathbf{x}_{e,m}$ (m wird als Index verwendet, da eventuell mehrere Punkte $\mathbf{x}_{e,m}$ existieren) mit $\dot{V}(\mathbf{x}_{e,m}) = 0$ Ruhelagen. Da $\dot{V}(\mathbf{x}) \leq 0$ ist, erhält man schließlich $\dot{V}(\mathbf{x}) < 0$ für $\mathbf{x} \neq \mathbf{x}_{e,m}$. Im Beispiel aus Abbildung 4.2 ist der Funktionswert in x_1 -Richtung in Punkt 2 positiv. Damit ergibt sich als obere Grenze $f_{\max,l,1}$ eben dieser positive Funktionswert. Als untere Grenze ergibt sich der Wert in der Ruhelage, also $f_{\min,l,1} = 0$. Entsprechendes gilt für die Komponente x_2 . Da $f_{\min,l,i}$ und $f_{\max,l,i}$ Ergebnisse einer numerischen Optimierung sind, ist Gl. (4.12) aus numerischen Gründen im Regelfall nur näherungsweise erfüllt. Deshalb überprüfen man in der Praxis $f_{\max,l,i} \approx f_{\min,l,i} \approx 0$.

Bemerkung 4.1. Der Verifikationsschritt kann zu numerischen Problemen führen, wenn Hyperquader untersucht werden, die keine Punkte enthalten, für die $\dot{V}(\mathbf{x}) = 0$ ist. Diese Probleme können vermieden werden, wenn zunächst nach den Hyperquadern gesucht wird, die Punkte mit $\dot{V}(\mathbf{x}) = 0$ enthalten. Diese Suche kann durch Lösen des SOS-Problems für $l = 1, \dots, K$

$\min w_{\max,l}$, sodass

$$w_{\max,l} - \frac{\partial V}{\partial \mathbf{x}} \mathbf{f}_l(\mathbf{x}) - \sum_{k=1}^{2n} b_{lk}(\mathbf{x}) r_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}], \quad (4.13)$$

$$b_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}], \quad k = 1, \dots, 2n,$$

mit den Hilfspolynomen $b_{lk}(\mathbf{x})$ erfolgen. Aus Satz 4.3 folgt, dass die Variable $w_{\max,l}$ eine obere Grenze von $\dot{V}(\mathbf{x})$ im Hyperquader \mathcal{X}_l ist. Schließt man alle Hyperquader mit $w_{\max,l} < -\epsilon$ mit kleinem $\epsilon > 0$ aus, so bleiben diejenigen Hyperquader, die Punkte mit $\dot{V}(\mathbf{x}) = 0$ enthalten.

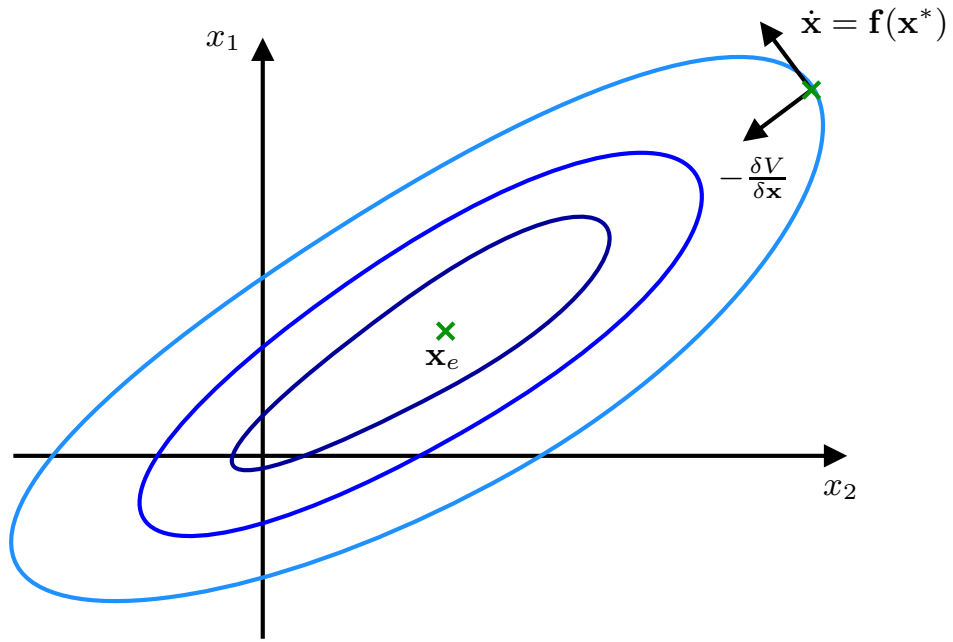


Abbildung 4.2: Höhenlinien einer möglichen Lyapunov-Funktion mit zugehöriger Ruhelage sowie einem weiteren Punkt \mathbf{x}^* , in dem $\dot{V}(\mathbf{x}^*) = 0$ gilt.

Schritt 2.2: Verifikation von $V(\mathbf{x}_e) < V(\mathbf{x})$.

Im zweiten Schritt wird nun die Bedingung $V(\mathbf{x}_e) < V(\mathbf{x})$ überprüft. Die Lösung besteht in der Überprüfung, ob das Polynom $V(\mathbf{x})$ nur genau ein Minimum in \mathbb{R}^n besitzt, sodass $V(\mathbf{x}_e) < V(\mathbf{x})$ erfüllt ist. Als Konsequenz aus diesem Lösungsansatz wird zunächst angenommen, dass nur eine Ruhelage \mathbf{x}_e existiert. Der Fall mehrerer Ruhelagen wird später näher beleuchtet.

Die Überprüfung, ob $V(\mathbf{x})$ nur ein Minimum in \mathbb{R}^n besitzt, kann wiederum in ein SOS-Problem überführt werden. Man berechnet obere und untere Grenzen von $V(\mathbf{x})$ in den Punkten mit $\frac{\partial V}{\partial \mathbf{x}} = \mathbf{0}$, also in den Extrem- und Sattelpunkten von $V(\mathbf{x})$, indem man

$$\begin{aligned} & \max v_{\min}, \text{ sodass} \\ & -v_{\min} + V(\mathbf{x}) - \sum_{k=1}^n \frac{\partial V}{\partial x_k} d_{k1}(\mathbf{x}) \in \Sigma[\mathbf{x}] \end{aligned} \quad (4.14)$$

und

$$\begin{aligned} & \min v_{\max}, \text{ sodass} \\ & v_{\max} - V(\mathbf{x}) - \sum_{k=1}^n \frac{\partial V}{\partial x_k} d_{k2}(\mathbf{x}) \in \Sigma[\mathbf{x}], \end{aligned} \quad (4.15)$$

mit den Hilfspolynomen $d_{ki}(\mathbf{x})$ löst. Außerdem ermittelt man eine untere Grenze von $V(\mathbf{x})$ auf dem Rand $\partial\mathcal{Y}$ der Menge $\mathcal{Y} \supset \mathcal{X}$, wobei $\partial\mathcal{Y} = \{\mathbf{x} \mid y(\mathbf{x}) = x_1^2 + \dots + x_n^2 = \beta\}$ mit ausreichend großem β gewählt wird. Das zugehörige Optimierungsproblem lautet

$$\begin{aligned} & \max v_{\partial\mathcal{Y},\min}, \text{ so dass} \\ & -v_{\partial\mathcal{Y},\min} + V(\mathbf{x}) - (y(\mathbf{x}) - \beta)h(\mathbf{x}) \in \Sigma[\mathbf{x}] \end{aligned} \quad (4.16)$$

mit dem Hilfspolynom $h(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$. Gilt nun

$$v_{\min} = v_{\max} < v_{\partial\mathcal{Y},\min}, \quad (4.17)$$

dann besitzt $V(\mathbf{x})$ genau ein Minimum in der Ruhelage \mathbf{x}_e . Abbildung 4.3 zeigt zwei Fälle, in denen obige Bedingungen nicht erfüllt sind. Die mögliche Lyapunov-Funktion in Abbildung 4.3(a) enthält mehrere Minima und Maxima. Wird nun mithilfe der SOS-Probleme 4.14 und 4.15 eine untere und obere Grenze für die Funktionswerte $V(\mathbf{x})$ in diesen Extrempunkten bestimmt, so ergeben sich unterschiedliche Werte für v_{\min} und v_{\max} , es liegen also mehrere Minima vor. Nur einen Extrempunkt besitzt die mögliche Lyapunov-Funktion in Abbildung 4.3(b). Allerdings liegt hier eine nach unten geöffnete Funktion vor. Um dieses auszuschließen wird der Funktionswert auf dem Rand $\partial\mathcal{Y}$ des Gebietes \mathcal{Y} untersucht. Für diesen gilt in diesem Fall $v_{\partial\mathcal{Y},\min} < v_{\min} = v_{\max}$, sodass Bedingung (4.17) nicht erfüllt ist.

Die zuvor hergeleiteten Ergebnisse sollen nochmal zusammengefasst werden.

Satz 4.5. *Sei \mathbf{x}_e eine unbekannte Ruhelage des hybriden Systems $\dot{\mathbf{x}} = \mathbf{f}_l(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}_l$, $l = 1, \dots, K$. Falls eine polynomiale Funktion $V(\mathbf{x})$ und Hilfspolynome $a_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}]$, $l = 1, \dots, K$ und $k = 1, \dots, 2n$, existieren, sodass*

$$\begin{aligned} & V(\mathbf{x}) \in \Sigma[\mathbf{x}], \\ & -\frac{\partial V}{\partial \mathbf{x}} \mathbf{f}_l(\mathbf{x}) - \sum_{k=1}^{2n} a_{lk}(\mathbf{x}) r_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}], \\ & f_{\max,l,i} = f_{\min,l,i} = 0, \quad l = 1, \dots, K, \quad i = 1, \dots, n, \end{aligned} \quad (4.18)$$

$$v_{\min} = v_{\max} < v_{\partial\mathcal{Y},\min}, \quad (4.19)$$

erfüllt ist, wobei $f_{\max,l,i}$, $f_{\min,l,i}$ und v_{\min} , v_{\max} , $v_{\partial\mathcal{Y},\min}$ Ergebnisse der SOS-Probleme (4.10)-(4.11) und (4.14)-(4.16) sind, dann ist die unbekannte Ruhelage \mathbf{x}_e lokal asymptotisch stabil.

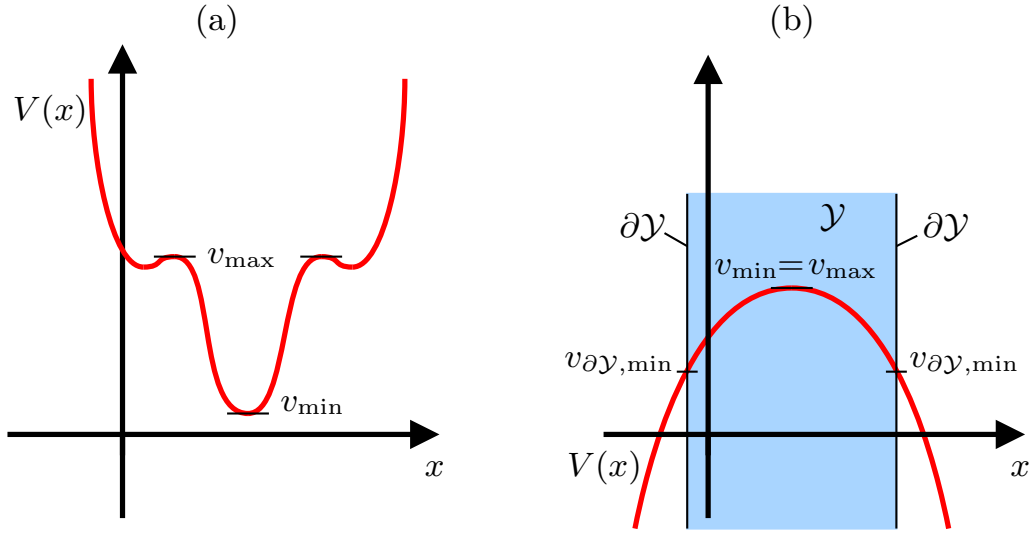


Abbildung 4.3: Beispiele für mögliche Lyapunov-Funktionen, die den Verifikationsschritt 2.2 nicht erfüllen: (a) Mögliche Lyapunov-Funktion mit mehreren lokalen Minima, sodass $v_{\min} \neq v_{\max}$. (b) Nach unten geöffnete Lyapunov-Funktion, bei der zwar $v_{\min} = v_{\max}$ ist, Bedingung (4.17) aber nicht erfüllt ist.

Bisher wurde nur die Stabilität von unbekannten Ruhelagen in KRFS untersucht. Die zuvor beschriebenen Schritte können aber mit wenigen Modifikationen auf den zeitdiskreten Fall übertragen werden. Prinzipiell muss zunächst $\dot{V}(\mathbf{x})$ durch $V(\mathbf{f}(\mathbf{x})) - V(\mathbf{x})$ ersetzt werden. Da nun zeitdiskrete RFS betrachtet werden, ist die Ruhelage durch $\mathbf{x}_e = \mathbf{f}(\mathbf{x}_e)$ gegeben. Entsprechend muss Bedingung (4.18) aus Schritt 2.1 durch (4.20)-(4.21) ersetzt werden, wodurch für sich der folgende Satz ergibt:

Satz 4.6. *Sei \mathbf{x}_e eine unbekannte Ruhelage des hybriden Systems $\mathbf{x}(k+1) = \mathbf{f}_l(\mathbf{x}(k))$, $\mathbf{x} \in \mathcal{X}_l$, $l = 1, \dots, K$. Falls eine polynomiale Funktion $V(\mathbf{x})$ und Hilfspolynome $a_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}]$, $l = 1, \dots, K$ und $k = 1, \dots, 2n$, existieren, so dass*

$$V(\mathbf{x}) \in \Sigma[\mathbf{x}],$$

$$V(\mathbf{x}(k)) - V(\mathbf{f}_l(\mathbf{x}(k))) - \sum_{k=1}^{2n} a_{lk}(\mathbf{x}) r_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}],$$

$$f_{\max,l,i} = f_{\min,l,i} := x_i^*, \quad l = 1, \dots, K, \quad i = 1, \dots, n, \quad (4.20)$$

$$\mathbf{x}^* = \mathbf{f}(\mathbf{x}^*), \quad (4.21)$$

$$v_{\min} = v_{\max} < v_{\partial\mathcal{Y},\min},$$

erfüllt ist, wobei $f_{\max,l,i}$, $f_{\min,l,i}$ Ergebnisse der Optimierungsprobleme

$$\begin{aligned} & \max f_{\min,l,i}, \text{ so dass} \\ & -f_{\min,l,i} + f_{l,i}(\mathbf{x}) + (V(\mathbf{x}(k)) - V(\mathbf{f}_l(\mathbf{x}(k))))c_{li1}(\mathbf{x}) \\ & - \sum_{k=1}^{2n} a_{lk}(\mathbf{x})r_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}], \end{aligned} \quad (4.22)$$

$$\begin{aligned} & \min f_{\max,l,i}, \text{ so dass} \\ & f_{\max,l,i} - f_{l,i}(\mathbf{x}) + (V(\mathbf{x}(k)) - V(\mathbf{f}_l(\mathbf{x}(k))))c_{li2}(\mathbf{x}) \\ & - \sum_{k=1}^{2n} a_{lk}(\mathbf{x})r_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}], \end{aligned} \quad (4.23)$$

mit $a_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}]$ und v_{\min} , v_{\max} , $v_{\partial\mathcal{Y},\min}$ Ergebnisse der SOS-Probleme (4.14)-(4.16) sind, dann ist die Ruhelage $\mathbf{x}_e = \mathbf{x}^*$ lokal asymptotisch stabil.

Bemerkung 4.2. Bei der Analyse von KRFS ist die zunächst unbekannte Ruhelage auch nach dem Nachweis der Stabilität unbekannt. Im Gegensatz dazu wird bei DRFS die zunächst unbekannte Ruhelage während der Stabilitätsanalyse durch die SOS-Probleme (4.22)-(4.23) bestimmt.

Die Gebiete

$$\mathcal{V}_c = \{\mathbf{x} \mid V(\mathbf{x}) \leq c\} \subseteq \mathcal{X}$$

sind hierbei Einzugsgebiete von \mathbf{x}_e . Alle Einzugsgebiete müssen Untermen-
gen von \mathcal{X} sein, da nur hier $\dot{V}(\mathbf{x}) < 0$ (bzw. $\Delta V(\mathbf{x}) < 0$) gesichert ist.

Bemerkung 4.3. Falls die Verifikationsschritte für eine bestimmte Lyapunov-Funktion nicht erfolgreich sind, kann zunächst die Ordnung der angesetzten Polynome, also des Lyapunov-Polynoms und der Hilfspolynome, erhöht werden. Eine andere Möglichkeit besteht darin, das für die Stabilitätsanalyse betrachtete Gebiet zu begrenzen und hierfür lokale Stabilität zu prüfen. Dies ist insbesondere dann notwendig, wenn mehrere Ruhelagen in einem Hyperquader existieren. Dies soll nun diskutiert werden.

Mehrere Ruhelagen

Im zweiten Verifikationsschritt wurde angenommen, dass nur eine Ruhelage existiert. Der Fall mehrerer Ruhelagen kann allerdings mit leichten Modifikationen berücksichtigt werden. Hierzu wird jede Ruhelage einzeln

mit einem umgebenden Gebiet betrachtet. Das Gebiet, in dem $\dot{V}(\mathbf{x}) < 0$ (bzw. $\Delta V(\mathbf{x}) < 0$) verlangt wird, wird also derart beschränkt, dass nur eine Ruhelage in diesem Gebiet liegt. Dies kann beispielsweise durch eine Verringerung der Zahl der betrachteten elementaren Hyperquader erfolgen. Eine andere Möglichkeit ist es, ein Polynom $g(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$, Hilfspolynome $a_{lk}(\mathbf{x}) \in \Sigma[\mathbf{x}]$, $l = 1, \dots, K$, $k = 1, \dots, 2n$, und $s(\mathbf{x}) \in \Sigma[\mathbf{x}]$ zu wählen, und Bedingung (4.9) in Schritt 1 durch

$$-\frac{\partial V}{\partial \mathbf{x}} \mathbf{f}_l(\mathbf{x}) - \sum_{k=1}^{2n} r_{lk}(\mathbf{x}) a_{lk}(\mathbf{x}) - g(\mathbf{x}) s(\mathbf{x}) \in \Sigma[\mathbf{x}], l = 1, \dots, K,$$

zu modifizieren. Dies verkleinert das untersuchte Gebiet auf das Gebiet $\mathcal{G} \cap \mathcal{X}$ mit $\mathcal{G} = \{\mathbf{x} \mid g(\mathbf{x}) \geq 0\}$. Entsprechend reduzieren sich die Einzugsgebiete auf

$$\mathcal{V}_c = \{\mathbf{x} \mid V(\mathbf{x}) \leq c\} \subseteq \mathcal{G} \cap \mathcal{X}.$$

Mithilfe dieser Modifikationen können die Ruhelagen nacheinander analysiert werden. Im zeitdiskreten Fall geht man analog vor.

Bemerkung 4.4. In allen SOS-Problemen muss die maximale Ordnung der unbestimmten Polynome, wie das Polynom V in Schritt 1 sowie alle Hilfspolynome $a_{lk}, b_{lk}, c_{li}, d_k, h, s$, ausreichend hoch angesetzt werden. Ein einfacher Ansatz ist, mit niedrigen Ordnungen zu beginnen und die Ordnung dann sukzessive zu erhöhen, bis zufriedenstellende Ergebnisse erreicht werden. Zur Lösung der SOS-Probleme existieren verschiedene Matlab Software Pakete wie YALMIP [121] oder SOSTOOLS [159], welche die SOS-Bedingungen in lineare Matrixungleichungen (LMI) umwandeln. Diese LMIs können dann mit gängigen Solvern wie SDTP3 [193] gelöst werden. Die Komplexität der zu lösenden SOS-Probleme lässt sich am Beispiel der SOS-Bedingungen (4.8)-(4.9) wie folgt bestimmen [149]: Wenn K die Anzahl der Hyperquader und n die Systemordnung ist, dann ergeben sich K SOS-Bedingungen (4.9) und $2nK$ Bedingungen für die Hilfspolynome. Somit sind insgesamt $K + 2nK + 1$ SOS-Probleme zu lösen. Bezeichnet man darüber hinaus $\deg(V)$ und $\deg(a_{lk})$ als Ordnungen der Lyapunov-Funktion und der Hilfspolynome, sind in den $K + 2nK + 1$ SOS-Problemen insgesamt $\binom{n+\deg(V)}{\deg(V)} + 2nK \binom{n+\deg(a_{lk})}{\deg(a_{lk})}$ freie Koeffizienten zu bestimmen. Entsprechend ist die Komplexität bei hohen Systemordnungen und hohen Ordnungen der Hilfspolynome sehr hoch. Die Komplexität der weiteren SOS-Probleme kann in ähnlicher Weise berechnet werden.

4.4 Beispiele

4.4.1 Zeitkontinuierliches System

Betrachtet wird ein KRFS mit zwei Zuständen. Die Regelbasis des Systems ist in Abbildung 4.4 zu sehen. Die Parameter sind Tabelle 4.1 zu entnehmen. Die Ruhelage wird als unbekannt angenommen.

		x_1			
x_2	\dot{x}_1/\dot{x}_2	n	nu	k	g
	n	pm/pg	nk/pg	nm/pm	nm/pk
	nu	pg/nk	pm/pk	pk/pm	ng/pg
	k	pm/pk	pm/pm	pk/nm	nm/nk
	g	pk/nm	pk/ng	pm/ng	ng/nm

Abbildung 4.4: Regelbasis des betrachteten KRFS mit n="negativ", nu="null", k="klein", g="groß", ng="negativ groß", nm="negativ mittel", nk="negativ klein", pk="positiv klein", pm="positiv mittel", pg="positiv groß".

Tabelle 4.1: Linguistische Werte der Regelbasis.

$s_{w_i}^{\dot{x}_i}$						$s_{j_i}^{x_i}$			
ng	nm	nk	pk	pm	pg	n	nu	k	g
-0,9	-0,5	-0,1	0,1	0,5	0,9	-1	0	1	2

Aus Schritt 1 ergibt sich als mögliche Lyapunov-Funktion

$$\begin{aligned}
 V(x_1, x_2) = & 540,3 - 459,19x_1 - 40,55x_2 - 137,46x_1x_2 + 2,91x_1^2 \\
 & + 40,71x_2^2 - 46,79x_1^3 - 7,1x_1^2x_2 + 2,99x_1x_2^2 - 33,93x_2^3 \\
 & + 111,87x_1^4 - 86,85x_1^3x_2 + 181,65x_1^2x_2^2 - 8,24x_1x_2^3 + 73,19x_2^4.
 \end{aligned}$$

Es soll nun überprüft werden, ob die polynomiale Funktion tatsächlich eine Lyapunov-Funktion ist. Durch Anwendung des SOS-Problems (4.13) ergibt sich, dass $\dot{V}(\mathbf{x}) = 0$ nur im mittleren rechten Hyperquader (vgl. Abbildung 4.5) mit dem Indexvektor $\mathbf{j}_l = [3 \ 2]^T$ auftritt. Entsprechend

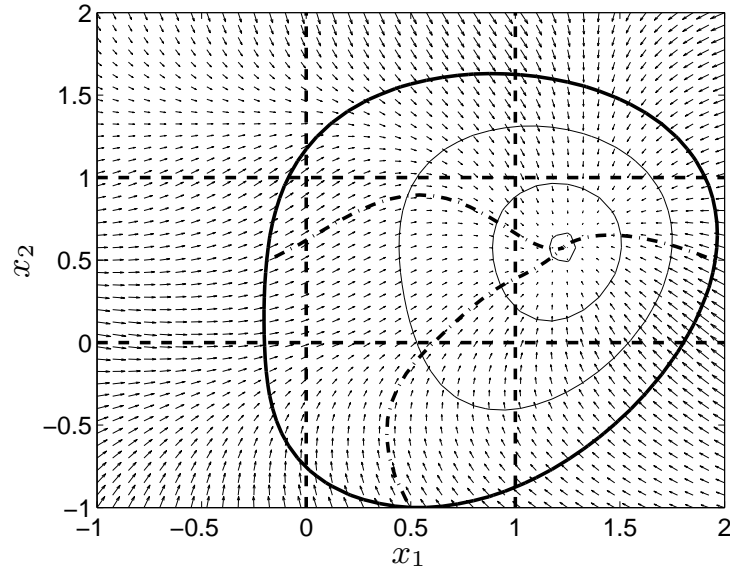


Abbildung 4.5: Trajektorien (strichpunktiert) des KRFS und Höhenlinien der konstruierten Lyapunov-Funktion (durchgezogen). Die gestrichelten Linien deuten die Grenzen der Hyperquader an.

kann sich die Analyse auf diesen Hyperquader beschränken. Die unteren und oberen Grenzen für das Vektorfeld dieses Hyperquaders für Punkte mit $\dot{V}(\mathbf{x}) = 0$ ergeben sich zu $f_{\min,6,1} = -1,25 \cdot 10^{-6}$, $f_{\max,6,1} = 9,99 \cdot 10^{-5}$, $f_{\min,6,2} = -1,9 \cdot 10^{-5}$ und $f_{\max,6,2} = 2,48 \cdot 10^{-6}$. Bedingung (4.18) ist damit erfüllt. Außerdem folgt aus Bedingung (4.19) $v_{\min} = v_{\max} = 34,43$ und mit $\beta = 2,5$, $v_{\partial\mathcal{Y},\min} = 67,77$.

Damit sind alle Bedingungen von Satz 4.5 erfüllt und das KRFS besitzt eine asymptotisch stabile Ruhelage im mittleren rechten Hyperquader. Abbildung 4.5 zeigt die Trajektorien des Systems zusammen mit den Höhenlinien der Lyapunov-Funktion. Die fett gedruckte Linie zeigt das berechnete Einzugsgebiet $\mathcal{V}_c \subset \mathcal{X}$.

4.4.2 Populationsdynamik

Als zweites Beispiel wird der Mutualismus aus Beispiel 3.1 in Abschnitt 3.1 betrachtet, diesmal allerdings als zeitdiskretes System. Die zugehörige Regelbasis des DRFS zeigt Abbildung 4.6.

Es wird wiederum der beschriebenen Ansatz angewendet, um asymptotische Stabilität nachzuweisen, ohne die Ruhelage exakt zu kennen. Um ein besser konditioniertes SOS-Problem zu erhalten, werden die Zustandsvariablen in das Intervall $[0, 1]$ transformiert. Dies ändert nichts an der Dynamik des Systems, sodass die Ergebnisse zur Stabilität des transformierten Systems auch für das Originalsystem gültig bleiben. Die norma-

		$x_1(k)$					$x_1(k)$		
		k	m	g			k	m	g
$x_2(k)$	$x_1(k+1)$	k	m	g	$x_2(k)$	$x_1(k+1)$	k	m	g
	k	m	k	m		k	m	mk	mk
	m	mk	g	m		m	k	g	mg
	g	mk	mg	mg		g	m	m	mg

Abbildung 4.6: Regelbasis des DRFS zur Beschreibung des Mutualismus mit k="klein", mk="mittel klein", m="mittel", mg="mittel groß", g="groß".

lisierten Werte der Ausgangskernpositionen lauten $s_1^{x_i} = 0$, $s_2^{x_i} = 0,25$, $s_3^{x_i} = 0,5$, $s_4^{x_i} = 0,75$ und $s_5^{x_i} = 1$ für $i = 1,2$.

Aus Schritt 1 ergibt sich das Polynom

$$\begin{aligned}
V(x_1, x_2) = & 12550 - 1709x_1 - 1701x_2 - 3944x_1x_2 + 3604x_1^2 \\
& + 3631x_2^2 + 1237x_1^3 - 1729x_1^2x_2 - 1696x_1x_2^2 + 1199x_2^3 \\
& + 251x_1^4 - 1207x_1^3x_2 + 1996x_1^2x_2^2 - 1204x_1x_2^3 + 248x_2^4.
\end{aligned}$$

Im zweiten Schritt zeigt das SOS-Problem (4.13) wiederum, dass $\dot{V}(\mathbf{x}) = 0$ nur im oberen rechten Hyperquader in Abbildung 4.7 auftritt. Die obere und untere Grenze des Vektorfelds in diesem Hyperquader ist $f_{\min,4,1} = f_{\max,4,1} = 0,75$ und $f_{\min,4,2} = f_{\max,4,2} = 0,75$. Da $\mathbf{f}([0,75 \ 0,75]^T) = [0,75 \ 0,75]^T$ ist, ist Bedingung (4.12) erfüllt. Des Weiteren erhält man für Bedingung (4.17) $v_{\min} = v_{\max} = 11454$ und $\beta = 5$ führt zu $v_{\partial\mathcal{Y},\min} = 12002$, womit alle Bedingungen aus Satz 4.6 erfüllt sind. Das DRFS hat damit im oberen rechten Hyperquader die asymptotisch stabile Ruhelage $\mathbf{x}_e = [0,75 \ 0,75]^T$. Abbildung 4.7 zeigt eine Trajektorie des DRFS und Höhenlinien der konstruierten Lyapunov-Funktion.

4.5 Zusammenfassung und Fazit

Dieses Kapitel beschäftigte sich mit der Stabilitätsanalyse rekurrenter Fuzzy-Systeme. Die Verwendung der hybriden polynomialen Systemdarstellung ermöglichte die Analyse von zeitdiskreten und zeitkontinuierlichen RFS in einem gemeinsamen Rahmen mithilfe der SOS-Methodik. Hierbei lassen sich sowohl bekannte als auch unbekannte Ruhelagen untersuchen. Während sich die in [49, 97] vorgestellten Methoden sehr eng an der Regelbasis orientierten, arbeitet der im Rahmen der Arbeit entwickelte Ansatz

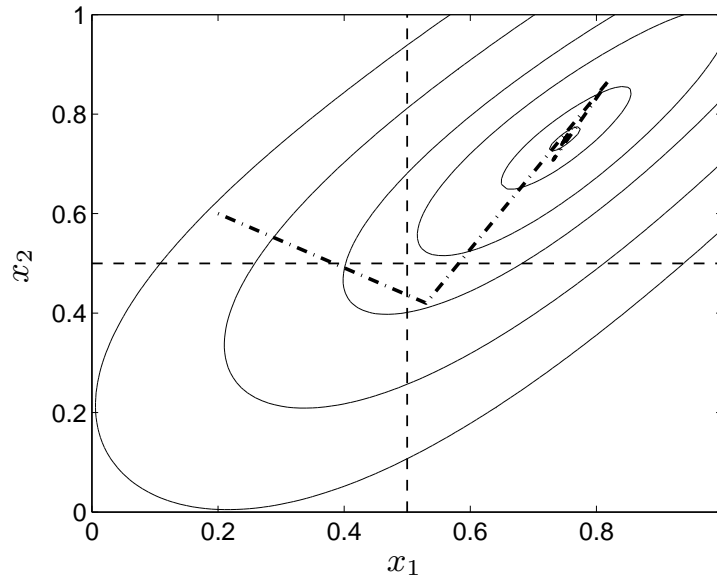


Abbildung 4.7: Trajektorie (strichpunktiert) des DRFS mit transformierten Zustandsvariablen und Höhenlinien der konstruierten Lyapunov-Funktion. Die gestrichelten Linien deuten die Grenzen der Hyperquader an.

ausschließlich mit der analytischen Darstellung RFS. Dies hat den Vorteil, dass die Analyse nicht auf spezielle Systemklassen bzw. Ruhelagen an speziellen Punkten eingeschränkt ist. Einschränkend muss allerdings darauf hingewiesen werden, dass die momentan verfügbaren Solver für SOS-Probleme bisher nur bei niedrigdimensionalen Systemen stabil funktionieren. In [194] wird von gelösten SOS-Problem für Systeme mit 8-10 Zuständen berichtet. Da im vorliegenden Fall hybride Systeme betrachtet werden, bei denen mehr Parameter verwendet werden, sind je nach Problem Systeme mit 6-7 Zuständen untersuchbar. Dies liegt insbesondere daran, dass die Zahl der Koeffizienten der Hilfspolynome mit steigender Systemdimension stark ansteigt, was sowohl numerische Probleme als auch Speicherprobleme hervorruft.

5 Modellierung mit zeitkontinuierlichen rekurrenten Fuzzy-Systemen

Eine vielbeachtete Anwendungsmöglichkeit der Fuzzy-Logik ist die Modellierung statischer und dynamischer Zusammenhänge. Hierbei eignen sich Fuzzy-Systeme insbesondere zur Modellierung sogenannter Grey-Box-Modelle, deren physikalische Beschreibung zwar nicht oder nur schwierig möglich ist, bei denen aber über reine Messdaten des Ein-/Ausgangsverhaltens hinaus qualitatives Wissen über das System vorhanden ist. Dieses Kapitel soll nun genauer beleuchten, wie dynamische Systeme durch rekurrente Fuzzy-Systeme nachgebildet werden können. Eine erste Anwendung zur Modellierung dynamischer Systeme wurde in [49] vorgestellt, bei der das elektrische Verhalten einer Brennstoffzelle mit zeitkontinuierlichen rekurrenten Fuzzy-Systemen nachgebildet wurde. Allerdings wurde die Modellbildung weitestgehend händisch durchgeführt. Um diesen Modellbildungsprozess zu automatisieren, sollen im Folgenden Verfahren entwickelt werden, mit denen das Verhalten des RFS an das zu modellierende System angepasst wird [6, 7, 172].

5.1 Einführung und Problemdefinition

Die Modellbildung mit rekurrenten Fuzzy-Systemen setzt sich grundsätzlich aus mehreren Teilproblemen zusammen. Erstens ist festzulegen, welche Prozessgrößen für die Modellierung relevant sind und entsprechend als Eingangs- oder Zustandsgrößen definiert werden. Zweitens muss die Struktur rekurrenter Fuzzy-Systeme, insbesondere die Zahl an Zugehörigkeitsfunktionen für jede Eingangs- und Zustandsgröße und die Regelbasis bestimmt werden. Drittens sind die freien Parameter des RFS festzulegen.

Zur Lösung der obigen Teilprobleme werden verschiedene Informationsquellen benötigt. Wie bereits angedeutet werden in dieser Arbeit insbesondere zwei Informationsquellen verwendet: Expertenwissen von Prozessan-

wendern und experimentell gewonnene Datensätze. Liegt ein System vor, bei dem dieses Wissen vorhanden ist, so lässt sich der Vorteil von Fuzzy-Systemen gegenüber reinen Black-Box Modellbildungswerkzeugen, nämlich die Möglichkeit, Expertenwissen in die Modellierung zu integrieren und mit Datensätzen zu kombinieren, besonders gut nutzen. Das qualitative Wissen über den Prozess kann abhängig vom Anwendungsfall verschiedene Ausprägungen und Detaillierungsgrade annehmen. In vielen Fällen sind die Eingänge und Zustände des Systems bekannt. Oftmals ist es auch möglich, Regeln über das Prozessverhalten und die Zusammenhänge zwischen den Eingangs- und Zustandsgrößen aufzustellen. Solche Regeln können dann in einer Regelbasis zusammengefasst werden. Dagegen ist die Festlegung der Parameter allein durch Expertenwissen nahezu unmöglich.

Entsprechend lassen sich abhängig vom Detaillierungsgrad des vorhandenen Prozesswissens einige Teilprobleme der Modellbildung nur datengestützt lösen. Hierzu werden Datensätze

$$(u_p^{(v)}(t), x_i^{(v)}(t), \dot{x}_i^{(v)}(t)) \quad \text{bzw.} \quad (u_p^{(v)}(k), x_i^{(v)}(k), x_i^{(v)}(k+1)) \quad (5.1)$$

mit $i = 1, \dots, n, p = 1, \dots, m, v = 1, \dots, N$ zur Optimierung der Dynamikfunktion \mathbf{f} benötigt. Die Daten sollen hierbei die Dynamik des Prozesses möglichst gut repräsentieren. Zur Optimierung der Ausgangsfunktion \mathbf{g} werden entsprechend Datensätze der Form

$$(u_p^{(v)}, x_i^{(v)}, y_i^{(v)}) \quad (5.2)$$

generiert.

Bemerkung 5.1. In den folgenden Abschnitten werden, entsprechend der üblichen Bezeichnung, die vom KRFS prädizierten Größen mit einem „Dach“ gekennzeichnet. Der hochgestellte Index (v) wie in (5.1)-(5.2) nummeriert die Datensätze.

Mit diesen Vorüberlegungen und Festlegungen kann nun das in den folgenden Abschnitten betrachtete Modellierungsproblem formuliert werden:

Modellierungsproblem 5.1. Gegeben sind Datensätze (5.1) bzw. (5.2) sowie verschiedene Formen von Vorwissen über das zu modellierende System. Ermittle auf Basis dieser Informationen

1. die relevanten Eingangs- und Zustandsgrößen des RFS,
2. die Struktur des RFS, d.h. die Zahl an Zugehörigkeitsfunktionen für jede Eingangs- und Zustandsgröße sowie die Regelbasis,

Tabelle 5.1: Übersicht über die Entsprechungen bei der Modellierung verschiedener Teile rekurrenter Fuzzy-Systeme.

	Datensätze	Parameter
Dynamikfunktion KRFS	$(u_p(t), x_i(t), \dot{x}_i(t))$	$s_j^x, s_q^u, s_{w(j,q)}^{\dot{x}}$
Ausgangsfunktion KRFS	$(u_p(t), x_i(t), y_i(t))$	$s_j^x, s_q^u, s_{v(j,q)}^y$
Dynamikfunktion DRFS	$(u_p(k), x_i(k), x_i(k+1))$	$s_j^x, s_q^u, s_{w(j,q)}^x$
Ausgangsfunktion DRFS	$(u_p(k), x_i(k), y_i(k))$	$s_j^x, s_q^u, s_{v(j,q)}^y$

3. optimale Werte der freien Parameter des RFS, also der Kernpositionen und Kernpositionsableitungen.

Im Folgenden sollen Algorithmen vorgestellt werden, mit denen die in Modellierungsproblem 5.1 beschriebenen Teilprobleme gelöst werden können. Die Algorithmen sind sowohl zur Modellierung dynamischer Systeme mit zeitkontinuierlichen als auch mit zeitdiskreten RFS ohne Rückkopplungskorrespondenz anwendbar. Dabei können die Struktur und die Parameter sowohl der Dynamikfunktion \mathbf{f} als auch der Ausgangsfunktion \mathbf{g} mit den Verfahren optimiert werden. Die weiteren Ausführungen werden sich auf die Modellierung der Dynamikfunktion zeitkontinuierlicher RFS konzentrieren. Tabelle 5.1 zeigt, welche Prozessgrößen bzw. Parameter bei der Modellierung der Ausgangsfunktion sowie bei zeitdiskreten RFS den Größen bzw. Parameter der betrachteten Dynamikfunktion zeitkontinuierlicher RFS entsprechen. Diese Größen und Parameter können dann, je nach Anwendungsfall, in den vorgestellten Algorithmen ersetzt werden. Zeitdiskrete RFS mit Rückkopplungskorrespondenz können dagegen mit den hier vorgestellten Verfahren nicht behandelt werden. Auf sie wird in Kapitel 6 näher eingegangen.

Das Kapitel ist derart aufgebaut, dass die vorgestellten Algorithmen zunehmend weniger Vorwissen benötigen. So stellt Abschnitt 5.2 zunächst die Optimierung der Parameter des RFS vor. Die Struktur des RFS wird hierbei als gegeben vorausgesetzt. Außerdem wird gezeigt, wie weitergehendes Vorwissen, z.B. in Form einer gegebenen Regelbasis oder Kenntnissen über das System in Bereichen des Zustandsraums, in die Parameteroptimierung integriert werden kann. In Abschnitt 5.3 wird dann ein Algorithmus vorgestellt, mit dem auch die Struktur des RFS ausschließlich datengestützt ermittelt wird. Im Ergebnis erhält man so eine vollständige Regelbasis auf

Basis der vorgegebenen Datensätze. Abschließend wird in Abschnitt 5.4 gezeigt, wie datengestützt die relevanten Eingangs- und Zustandsgrößen ausgewählt werden können.

5.2 Parameteroptimierung

Zunächst wird der Ansatz zur Parameteroptimierung, also der Lösung von Teilproblem 3 des Modellierungsproblem 5.1, vorgestellt. Hierfür wird angenommen, dass die Struktur des KRFS, d.h. die Eingänge und Zustände sowie die Zahl der Zugehörigkeitsfunktionen für jede Eingangs- und Zustandsgröße und Zustandsableitung, zuvor festgelegt wurde. Dies kann entweder durch qualitatives Wissen über die Systemdynamik in Form von Wenn-dann-Regeln oder durch eine Strukturoptimierung, wie in Abschnitt 5.3 beschrieben, erfolgen.

Prinzipiell wird bei KRFS zwischen zwei Parametertypen unterschieden, nämlich den Kernpositionsableitungen $s_{w_i(\mathbf{j}, \mathbf{q})}^{\dot{x}_i}$, die die Ausgangszugehörigkeitsfunktionen definieren und den Kernpositionen $s_{j_i}^{x_i}$ und $s_{q_p}^{u_p}$, die die Zugehörigkeitsfunktionen der Eingänge und Zustände definieren. Die Veränderungen, die die beiden Parametergruppen am KRFS vornehmen, sind in Abbildung 5.1 verdeutlicht. Die Veränderung einer Kernposition führt anschaulich zu einer Verschiebung des Gitters im Eingangs-/Zustandsraum \mathcal{Z} . Wie bereits in Abschnitt 3.1 angedeutet, geben die Kernpositionsableitungen den Verlauf der Systemtrajektorie in der zugehörigen Kernposition an. Durch Veränderung einer Kernpositionsableitung wird entsprechend ihre Länge und Orientierung angepasst.

Im Folgenden wird zunächst die Optimierung der Kernpositionsableitungen unter Nutzung verschiedener Formen von Vorwissen beschrieben. Anschließend wird auf die Optimierung der Kernpositionen genauer eingegangen.

5.2.1 Optimierung der Kernpositionsableitungen

Es werden zur Optimierung der Kernpositionsableitungen vier Fälle, die unterschiedliche Anforderungen an das vorhandene Vorwissen stellen, unterschieden:

- Optimierung ohne Beschränkungen: Jeder Regel wird eine eigene Kernpositionsableitung $s_{w_i(\mathbf{j}, \mathbf{q})}^{\dot{x}_i}$ zugeordnet (Abbildung 5.2 links). Vorwissen wird hier nur für die Festlegung der Zahl der Eingangszugehörigkeitsfunktionen benötigt, weitergehendes Wissen ist dagegen

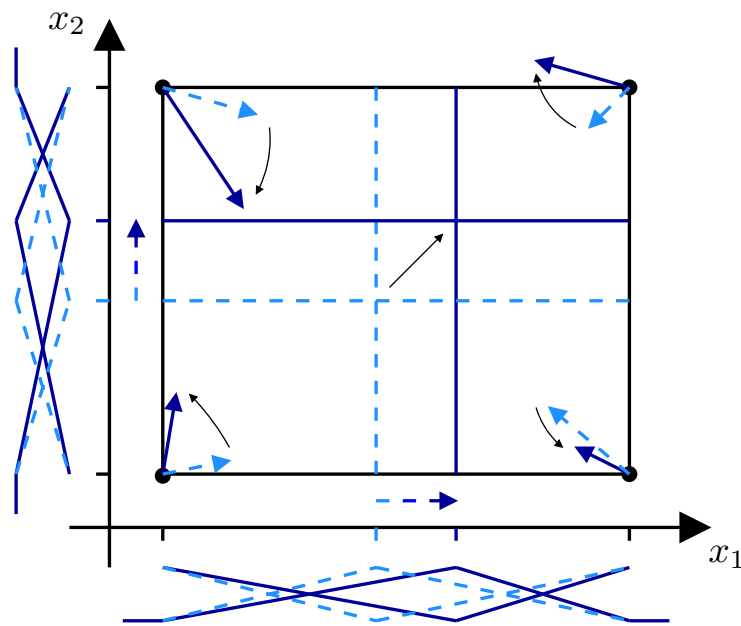


Abbildung 5.1: Anschauliche Bedeutung der Optimierung von Kernpositionen und Kernpositionsableitungen.

nicht erforderlich. Nachteilig ist allerdings die große Anzahl von Kernpositionsableitungen, durch die ihre linguistische Bedeutung verloren geht.

- Optimierung bei vollständiger Regelbasis: Die Zahl der linguistischen Werte der Kernpositionsableitungen wird eingeschränkt. Dies setzt voraus, dass anhand von Expertenwissen bereits die Struktur, d.h. eine Regelbasis, festgelegt wurde (Abbildung 5.2 rechts), wodurch die linguistische Interpretierbarkeit des KRFS gegeben ist. Während der Optimierung sind dann Nebenbedingungen einzuhalten.
- Optimierung zur flexiblen Partitionierung: Die Zahl der Kernpositionsableitungen wird wiederum eingeschränkt, in diesem Fall aber unter dem Fokus, eine flexiblere Partitionierung des Eingangsraums zu erhalten. Durch geeignete Nebenbedingungen können so auch von der gitterähnlichen Partitionierung abweichende Partitionierungen erzeugt werden.
- Optimierung bei Vorwissen über $\mathbf{f}(\mathbf{x}, \mathbf{u})$: Dieser Fall berücksichtigt Formen von Vorwissen, die die Dynamikfunktion direkt betreffen, beispielsweise die Ausprägung der Nichtlinearität der Dynamikfunktion in bestimmten Bereichen des Zustandsraums oder der Einfluss

		x_2						x_2			
\dot{x}_1		n	ns	ps	p	\dot{x}_1		n	ns	ps	p
	n	1	2	3	4		n	1	2	3	5
	z	5	6	7	8		z	2	1	2	3
	p	9	10	11	12		p	4	1	2	4

Abbildung 5.2: Regelbasis ohne Beschränkung (links) sowie vollständig definierte Regelbasis (rechts) für die Zustandsableitung \dot{x}_1 . Die Zahlen von 1-12 kennzeichnen jeweils unterschiedliche linguistische Werte der Zustandsableitungen $L_{\mathbf{w}(\mathbf{j},\mathbf{q})}^{\dot{x}_1}$.

verschiedener Eingänge und Zustände auf eine bestimmte Zustandsableitung.

Optimierung ohne Beschränkungen

Zur Herleitung wird zunächst die allgemeine Systemdarstellung rekurrenter Fuzzy-Systeme (3.5) in die Komponenten der Zustandsableitung aufgespalten:

$$\begin{aligned}
 \hat{\dot{x}}_i &= \sum_{\mathbf{j},\mathbf{q}} s_{w_i(\mathbf{j},\mathbf{q})}^{\dot{x}_i} \prod_{l=1}^n \mu_{j_l}^{x_l}(x_l) \prod_{p=1}^m \mu_{q_p}^{u_p}(u_p) \\
 &= \sum_{\mathbf{j},\mathbf{q}} \Xi_{(\mathbf{j},\mathbf{q})}(\mathbf{x},\mathbf{u}) \cdot s_{w_i(\mathbf{j},\mathbf{q})}^{\dot{x}_i} = \Xi^T(\mathbf{x},\mathbf{u}) \cdot \mathbf{s}^{\dot{x}_i},
 \end{aligned} \tag{5.3}$$

wobei im letzten Schritt die Summe über alle Regeln durch das Vektorprodukt dargestellt wird. Im Vektor $\Xi(\mathbf{x},\mathbf{u})$ sind die Prämissen (3.4) aller Regeln zusammengefasst. Der Vektor $\mathbf{s}^{\dot{x}_i}$ enthält die der jeweiligen Prämisse (Regel) zugeordneten Kernpositionsableitungen in x_i -Richtung. Somit enthält $\mathbf{s}^{\dot{x}_i}$ alle Kernpositionsableitungen der linguistischen Zustandsableitung \dot{x}_i .

Offensichtlich ist Gleichung (5.3) linear in den zu optimierenden Parametern $\mathbf{s}^{\dot{x}_i}$, so dass sich das Verfahren der kleinsten Quadrate (Least-Squares) zur Lösung eignet. Man minimiert also den quadratischen Fehler zwischen

gemessenem \dot{x}_i und berechnetem Ausgang $\hat{\dot{x}}_i$ der N Datenpaare:

$$\begin{aligned} J_i(\mathbf{s}^{\dot{x}_i}) &= \frac{1}{2} \sum_{v=1}^N \left(e_i^{(v)} \right)^2 = \frac{1}{2} \sum_{v=1}^N \left(\dot{x}_i^{(v)} - \hat{\dot{x}}_i^{(v)} \right)^2 \\ &= \frac{1}{2} \sum_{v=1}^N \left(\dot{x}_i^{(v)} - \Xi^T(\mathbf{x}^{(v)}, \mathbf{u}^{(v)}) \cdot \mathbf{s}^{\dot{x}_i} \right)^2. \end{aligned} \quad (5.4)$$

Insgesamt ergeben sich $i = 1, \dots, n$ konvexe Optimierungsprobleme, die unabhängig voneinander gelöst werden können.

Zur Vereinfachung der Schreibweise werden die Regressionsmatrix Ψ und Vektoren \mathbf{z}_i

$$\Psi = \begin{pmatrix} \Xi^T(\mathbf{x}^{(1)}, \mathbf{u}^{(1)}) \\ \vdots \\ \Xi^T(\mathbf{x}^{(N)}, \mathbf{u}^{(N)}) \end{pmatrix}, \quad \mathbf{z}_i = \begin{pmatrix} \dot{x}_i^{(1)} \\ \vdots \\ \dot{x}_i^{(N)} \end{pmatrix} \quad (5.5)$$

eingeführt. Mittels dieser Notation ergibt sich aus Gl. (5.4):

Optimierungsproblem 5.1.

$$\min_{\mathbf{s}^{\dot{x}_i}} \left(\mathbf{z}_i - \Psi \cdot \mathbf{s}^{\dot{x}_i} \right)^T \left(\mathbf{z}_i - \Psi \cdot \mathbf{s}^{\dot{x}_i} \right). \quad (5.6)$$

Die Lösung dieses Optimierungsproblems in der i -ten Komponente lautet [78]

$$\mathbf{s}^{\dot{x}_i} = [\Psi^T \cdot \Psi]^{-1} \Psi^T \mathbf{z}_i. \quad (5.7)$$

Im Regelfall ist allerdings die Matrixinversion in Gl. (5.7) schwierig, da die Matrix oftmals schlecht konditioniert ist. Ein effektives Verfahren zur Umgehung der entstehenden numerischen Probleme ist, mit der zu invertierenden Matrix eine Singulärwertzerlegung durchzuführen (s. Anhang A.1) [63, 201]. Damit werden nur noch die linear unabhängigen Spalten der Matrix zur Berechnung der Inversen verwendet.

Optimierung bei vollständiger Regelbasis

Ein Nachteil des im vorigen Abschnitt beschriebenen Vorgehens ist, dass die Interpretierbarkeit des KRFS verloren geht, da jeder Regel eine eigene Kernpositionsableitung zugeordnet wird. Da die Zahl der Regeln R exponentiell mit der Dimension $n + m$ des Fuzzy-Systems und der Zahl an

Zugehörigkeitsfunktionen in jeder Dimension $r_x(i)$ bzw. $r_u(p)$ gemäß

$$R = \prod_{i=1}^n r_x(i) \prod_{p=1}^m r_u(p) \quad (5.8)$$

ansteigt, wächst auch die Zahl der Kernpositionsableitungen exponentiell.

Will man die Interpretierbarkeit erhalten, so muss man die Zahl der linguistischen Werte der Kernpositionsableitungen begrenzen. Wie bereits erwähnt, muss hierzu die Regelbasis mithilfe qualitativen Wissens über das Systemverhalten festgelegt werden. Es muss also einer bestimmten Prämisse eine Konklusion aus der Menge der möglichen Konklusionen zugeordnet werden. Im Umkehrschluss bedeutet dies, dass einer Kernpositionsableitung mehrere Regeln zugeordnet sind, was in der folgenden Definition festgehalten ist:

Definition 5.1. Die Menge der zur Kernpositionsableitung $s_{w^*}^{\dot{x}_i}$ gehörenden Regeln ist:

$$\Lambda_i(s_{w^*}^{\dot{x}_i}) = \{(\mathbf{j}, \mathbf{q}) | s_{w_i(\mathbf{j}, \mathbf{q})}^{\dot{x}_i} = s_{w^*}^{\dot{x}_i}\}.$$

Die Menge $\Lambda_i(s_{w^*}^{\dot{x}_i})$ enthält alle Indexvektoren der Prämisse (\mathbf{j}, \mathbf{q}) , deren Konklusion $w_i(\mathbf{j}, \mathbf{q})$ durch den Index w^* gegeben ist, d.h., $\Lambda_i(s_{w^*}^{\dot{x}_i})$ beschreibt alle Regeln der Form (3.1), die denselben linguistischen Ausgangswert besitzen.

Um auch für diesen Fall eine Optimierungsaufgabe zu entwickeln, geht man wieder von der Systembeschreibung (5.3) aus. Allerdings sind die Vektoren $\Xi(\mathbf{x}, \mathbf{u})$ und $\mathbf{s}^{\dot{x}_i}$ von denen in Abschnitt 5.2.1 verschieden. Aufgrund der eingeschränkten Zahl linguistischer Werte gilt jetzt

$$\mathbf{s}^{\dot{x}_i} = [s_1^{\dot{x}_i} \quad \dots \quad s_{m_i}^{\dot{x}_i}]^T, \quad (5.9)$$

wobei m_i der Anzahl der linguistischen Werte der i -ten Zustandsableitung entspricht und die Kernpositionsableitungen aufsteigend bzgl. ihrer linguistischen Werte geordnet sind. Der Vektor $\Xi_i(\mathbf{x}, \mathbf{u})$ enthält wiederum die zur jeweiligen Konklusion gehörenden Prämissen und ist definiert durch

$$\Xi_i(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} \sum_{(\mathbf{j}, \mathbf{q}) \in \Lambda_i(s_1^{\dot{x}_i})} \Xi(\mathbf{j}, \mathbf{q})(\mathbf{x}, \mathbf{u}) \\ \vdots \\ \sum_{(\mathbf{j}, \mathbf{q}) \in \Lambda_i(s_{m_i}^{\dot{x}_i})} \Xi(\mathbf{j}, \mathbf{q})(\mathbf{x}, \mathbf{u}) \end{pmatrix} \quad (5.10)$$

mit $\Lambda_i(s_{w^*}^{x_i})$ nach Definition 5.1. Im Gegensatz zum unbeschränkten Fall ist $\Xi_i(\mathbf{x}, \mathbf{u})$ nun von der i -ten Komponente des Zustandsvektors abhängig, da sich die Beschränkungen in jeder Komponente \dot{x}_i unterscheiden können.

Mit (5.9), (5.10) und ähnlicher Zuordnung wie in (5.5) kann wiederum das Least-Squares-Problem (5.6) formuliert werden. Allerdings entstehen Nebenbedingungen, die bei der Optimierung eingehalten werden müssen. Logischerweise soll ein linguistischer Wert "niedrig" nicht größer als ein linguistischer Wert "hoch" sein. Es muss also

$$s_v^{\dot{x}_i} < s_{v+1}^{\dot{x}_i} \quad \forall v = 1, \dots, m-1.$$

gelten. Fasst man diese Bedingungen zusammen, so folgt

$$\mathbf{A} \cdot \mathbf{s}^{\dot{x}_i} < \mathbf{0}$$

mit

$$\mathbf{A} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(m-1) \times m}.$$

Insgesamt ergeben sich damit $i = 1, \dots, n$ konvexe quadratische Optimierungsprobleme der Gestalt

Optimierungsproblem 5.2.

$$\min_{\mathbf{s}^{\dot{x}_i}} (\mathbf{z}_i - \Psi \cdot \mathbf{s}^{\dot{x}_i})^T (\mathbf{z}_i - \Psi \cdot \mathbf{s}^{\dot{x}_i}),$$

$$\text{sodass } \mathbf{A} \cdot \mathbf{s}^{\dot{x}_i} < \mathbf{0},$$

Zur Lösung können effiziente Verfahren der quadratischen Programmierung (QP) genutzt werden [123].

Optimierung zur flexiblen Partitionierung

Wie bereits im einleitenden Kapitel 2 beschrieben, weisen RFS aufgrund ihrer Definition eine gitterähnliche Partitionierung des Eingangsraums auf. Dies hat Vorteile bei der Low-Level-Interpretierbarkeit, führt aber auch zu einem exponentiellen Anstieg der Regelanzahl, was die Lesbarkeit verschlechtert. Außerdem kann diese strikte Partitionierung dazu führen, dass mehr Regeln als notwendig erzeugt werden. Hierzu betrachten wir die in

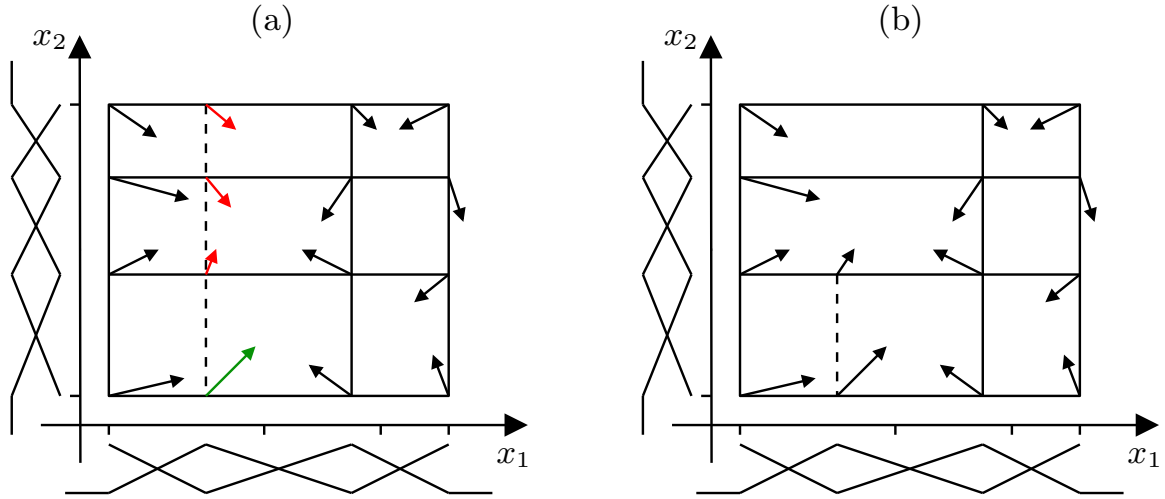


Abbildung 5.3: (a) Partitionierung bei Einfügen einer Kernposition in x_1 bei Optimierung ohne Nebenbedingungen. (b) Partitionierung bei Optimierung mit entsprechenden Nebenbedingungen.

Abbildung 5.3 (a) gezeigte Partitionierung und im Besonderen die strichliert eingezeichnete Kernposition in der Dimension x_1 . Durch Einfügen dieser Kernposition ergeben sich insgesamt vier zusätzliche Regeln und damit vier zusätzliche Kernpositionsableitungen, die in Abbildung 5.3 (a) grün und rot gekennzeichnet sind. Eventuell sind aber zur Modellierung nur ein Teil dieser neu hinzugekommenen Regeln notwendig. In dem betrachteten Beispiel nehmen wir an, dass das System im unteren linken Bereich schlecht nachgebildet wurde, sodass eine weitere Regel, deren Kernpositionsableitung grün markiert wurde, nötig ist. Die drei rot gekennzeichneten Kernpositionen sind dagegen zur Modellierung nicht vonnöten.

Im Folgenden soll nun gezeigt werden, wie diese Regeln durch Definition geeigneter Nebenbedingungen aus der Regelbasis entfernt und damit der exponentielle Anstieg der Regelanzahl vermieden werden kann. Hierzu gehen wir zunächst von der mathematischen Beschreibung der Zustandsableitung $\dot{\mathbf{x}}_s$ für einen beliebigen Punkt $(\mathbf{x}_s, \mathbf{u}_s)$ im l -ten Hyperquader mit Indexvektor $\mathbf{v}_l = (\mathbf{j}_l, \mathbf{q}_l)$ aus, die sich gemäß

$$\dot{\mathbf{x}}_s = \sum_{\mathbf{v} \in \{\mathbf{v}_l, \mathbf{v}_l + \mathbf{1}\}} \mathbf{s}_{\mathbf{w}(\mathbf{v})}^{\dot{\mathbf{x}}} \prod_{i=1}^n \mu_{j_i}^{x_i}(x_{i,s}) \prod_{p=1}^m \mu_{q_p}^{u_p}(u_{p,s})$$

aus der Interpolation der dem Hyperquader zugehörigen Kernpositionsableitungen ergibt. Wählt man nun die vier in Abbildung 5.3 gekennzeichneten neu hinzugekommenen Kernpositionsableitungen derart, dass sie sich

gemäß

$$\dot{\mathbf{s}}_{\mathbf{w},\text{interp}} = \sum_{\mathbf{v} \in \{\mathbf{v}_l, \mathbf{v}_l + \mathbf{1}\}} \dot{\mathbf{s}}_{\mathbf{w}(\mathbf{v})} \prod_{i=1}^n \mu_{j_i}^{x_i}(s_{j_i}^{x_i}) \prod_{p=1}^m \mu_{q_p}^{u_p}(s_{q_p}^{u_p}) \quad (5.11)$$

aus der Interpolation der benachbarten, schon vorhandenen Kernpositionsableitungen ergeben, so kann man die der Kernpositionsableitung zugeordnete Regel ohne Weiteres aus der Regelbasis entfernen, da sie redundant ist. Hierzu müssen die Eingangszugehörigkeitsfunktionen nicht angepasst werden. Es ergibt sich schließlich eine Partitionierung wie in Abbildung 5.3 gezeigt. Spaltet man nun Gl. (5.11) in seine Komponenten auf und löst auf, so erhält man

$$s_{w_i,\text{interp}}^{\dot{x}_i} - \sum_{\mathbf{v} \in \{\mathbf{v}_l, \mathbf{v}_l + \mathbf{1}\}} s_{w_i(\mathbf{v})}^{\dot{x}_i} \prod_{i=1}^n \mu_{j_i}^{x_i}(s_{j_i}^{x_i}) \prod_{p=1}^m \mu_{q_p}^{u_p}(s_{q_p}^{u_p}) = \mathbf{a}_{r,\text{itp}}^T \dot{\mathbf{s}}^{\dot{x}_i} = 0.$$

Definiert man abschließend die Matrix

$$\mathbf{A}_{\text{itp}}^T = \begin{pmatrix} \mathbf{a}_{1,\text{itp}} & \dots & \mathbf{a}_{r,\text{itp}} \end{pmatrix}, \quad (5.12)$$

die die Bedingungen (5.12) für alle redundanten Regeln zusammenfasst, so ergibt sich

Optimierungsproblem 5.3.

$$\min_{\dot{\mathbf{s}}^{\dot{x}_i}} (\mathbf{z}_i - \Psi \cdot \dot{\mathbf{s}}^{\dot{x}_i})^T (\mathbf{z}_i - \Psi \cdot \dot{\mathbf{s}}^{\dot{x}_i}),$$

sodass $\mathbf{A}_{\text{itp}} \cdot \dot{\mathbf{s}}^{\dot{x}_i} = \mathbf{0},$

Dies ist wiederum ein quadratisches Optimierungsproblem mit linearen Gleichungsbedingungen für die i -te Komponente des Zustandsvektors, dass mit Standardverfahren gelöst werden kann. Zu beachten ist, dass die Matrix \mathbf{A}_{itp} für jede Zustandsdimension identisch ist.

Optimierung bei Vorwissen über die Systemdynamik

Neben den im vorigen Abschnitt beschriebenen Einschränkungen, die durch die Definition einer vollständigen Regelbasis vor oder während der Optimierung entstehen, können sich auch aus anderen Formen von Vorwissen Einschränkungen bei der Optimierung der Kernpositionsableitungen ergeben [172].

Um die Grundidee zu erläutern, wird als einfaches Beispiel ein KRFS mit zwei Zuständen und ohne Eingänge betrachtet. Für einen elementaren Hyperquader ergibt sich nun die Differentialgleichung für den ersten Zustand nach Gl. (3.15) zu

$$\dot{x}_1 = \vartheta_0 + \vartheta_1 x_1 + \vartheta_2 x_2 + \vartheta_{12} x_1 x_2. \quad (5.13)$$

Fordert man nun, dass die Koeffizienten ϑ_0 und ϑ_{12} null sein sollen, erhält man offenbar aus dem ursprünglich multilinearen System ein lineares für den ersten Zustand. Fordert man dagegen, dass ϑ_2 und ϑ_{12} null sind, so erhält man ein affin-lineares System, das nur noch von x_1 , nicht aber von x_2 abhängt. Offensichtlich können also durch Nullsetzen einzelner Koeffizienten ϑ_i Systeme verschiedener Komplexität in verschiedenen elementaren Hyperquaden modelliert werden. Darüber hinaus können die Beschränkungen wieder für jede Zustandsdimension unabhängig definiert werden. In anderen Worten kann also mit diesem Ansatz für jede Zustandsdimension und jeden elementaren Hyperquader die Systemkomplexität durch Beschränkungen an die Koeffizienten eingestellt werden.

Diese Grundidee wird nun auf allgemeine KRFS generalisiert. Hierfür wird die Repräsentation eines elementaren Hyperquaders (3.15) in der i -ten Komponente

$$\dot{x}_i = \vartheta_l^{x_i} \omega,$$

betrachtet, wobei $\vartheta_l^{x_i}$ der i -te Zeilenvektor der Matrix Θ_l ist, der die Koeffizienten der i -ten Zustandsdifferentialgleichung enthält und ω wiederum den Monomvektor darstellt. Die Formulierung der Beschränkungen für die Koeffizienten $\vartheta_l^{x_i}$ der i -ten Zustandsdifferentialgleichung im l -ten Hyperquader kann durch die Definition einer Auswahlmatrix $\mathbf{B}_l^{x_i}$ erfolgen, so dass gilt:

$$\vartheta_l^{x_i} \mathbf{B}_l^{x_i} = \mathbf{0}^T.$$

Soll beispielsweise aus Gl. (5.13) ein lineares System entstehen, bildet man

$$\vartheta_l^{x_1} \mathbf{B}_l^{x_1} = \begin{bmatrix} \vartheta_0 & \vartheta_1 & \vartheta_2 & \vartheta_{12} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{0}^T.$$

Im Allgemeinen setzt sich das RFS aus mehr als einem Hyperquader zusammen. In diesem Fall können die für jeden Hyperquader unabhängig

festgelegten Beschränkungen in einer gemeinsamen Beschränkung

$$\boldsymbol{\vartheta}^{x_i} \mathbf{B}^{x_i} = \mathbf{0}^T \quad (5.14)$$

zusammengefasst werden, wobei $\boldsymbol{\vartheta}^{x_i} = [\boldsymbol{\vartheta}_1^{x_i} \ \dots \ \boldsymbol{\vartheta}_K^{x_i}]$ und

$$\mathbf{B}^{x_i} = \begin{bmatrix} \mathbf{B}_1^{x_i} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2^{x_i} & \dots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{B}_K^{x_i} \end{bmatrix}$$

gilt.

Bis hierhin sind die Beschränkungen (5.14) zur Einstellung der Systemkomplexität bezüglich der Koeffizienten definiert. Um diese nun in das Optimierungsproblem (5.6) nach Abschnitt 5.2.1 einbetten zu können, müssen die Beschränkungen in Abhängigkeit der Kernpositionsableitungen $\mathbf{s}^{\dot{x}_i}$ dargestellt werden. Dies ist allerdings sehr einfach möglich, da zwischen den Koeffizienten und den Kernpositionsableitungen durch Gl. (3.16) ein linearer Zusammenhang gegeben ist. Somit können die Bedingungen durch Einsetzen der i -ten Zeile von (3.16) in (5.14) in

$$\mathbf{s}^{\dot{x}_i} \boldsymbol{\Gamma} \mathbf{B}^{x_i} = \mathbf{0}^T$$

transformiert werden.

Die so definierten Beschränkungen können nun mit Optimierungsproblem (5.6) wie folgt kombiniert werden:

Optimierungsproblem 5.4.

$$\min_{\mathbf{s}^{\dot{x}_i}} (\mathbf{z}_i - \boldsymbol{\Psi} \cdot \mathbf{s}^{\dot{x}_i})^T (\mathbf{z}_i - \boldsymbol{\Psi} \cdot \mathbf{s}^{\dot{x}_i}),$$

$$\text{sodass } \mathbf{s}^{\dot{x}_i T} \boldsymbol{\Gamma} \mathbf{B}^{x_i} = \mathbf{0}^T.$$

Das Optimierungsproblem 5.4 stellt ein Least-Squares-Problem mit linearen Gleichungsnebenbedingungen dar und ist somit ein konvexes Optimierungsproblem. Wie auch im unbeschränkten Fall (5.7) wird das Problem 5.4 für jede Zustandsdimension definiert und kann unabhängig gelöst werden.

Das so erweiterte Optimierungsproblem erlaubt nun die Berücksichtigung verschiedener Formen von Vorwissen:

- *Abschnittsweise lineare Systeme*: Ist bekannt, dass es sich um ein abschnittsweise lineares System handelt, kann dies durch passende Nebenbedingungen berücksichtigt werden.
- *Spezielle nichtlineare Systeme*: Wie im vorigen Fall, kann auch Vorwissen über spezielle nichtlineare Systeme, beispielsweise bilineare oder eingangs-affine Systeme, eingebunden werden.
- *Systemverhalten in Unterräumen des Zustandsraums*: Da für jeden Hyperquader unterschiedliche Bedingungen formuliert werden können, kann Vorwissen über das Systemverhalten in Teilbereichen des Zustandsraumes eingebracht werden.
- *Zustands- und Eingangsgrößenauswahl*: Ist bekannt, dass eine Zustandsableitung nicht direkt durch einzelne Zustands- oder Eingangsgrößen beeinflusst wird, können diese Größen durch entsprechende Bedingungen ausgeschlossen werden.

Typische Prozesse, bei denen solches Vorwissen vorhanden ist, finden sich in der chemischen Industrie. Hier sind oftmals qualitative Informationen über die beteiligten Stoffe in Form von Massenbilanzen vorhanden. Dagegen sind die Reaktionskinetiken meist unbekannt [131]. Ein solcher Prozess wird in Abschnitt 5.5.2 genauer betrachtet.

Neben der Möglichkeit, Vorwissen einzubringen und damit die Interpretierbarkeit des RFS zu erhöhen, haben die obigen Ansätze zur Optimierung mit Gleichungsnebenbedingungen (Optimierungsproblem 5.3 und 5.4) noch einen weiteren Vorteil. Sie ermöglichen die Verringerung von Überanpassung des RFS an die Datensätze. Wie schon in Gl. (5.8) beschrieben, wächst die Zahl an Optimierungsparametern exponentiell mit der Systemordnung. Dies führt bei Systemen höherer Ordnung bereits bei der Verwendung einer relativ niedrigen Zahl an Zugehörigkeitsfunktionen pro Dimension zu einer sehr hohen Zahl an Parametern. Dadurch entsteht eine Systemstruktur, die komplexer ist, als es zur Modellbildung notwendig ist. Definiert man nun sinnvolle Nebenbedingungen, so wird die Zahl an freien Optimierungsparametern mit jeder Nebenbedingung um eins reduziert. Da viele der oben angegebenen Möglichkeiten, Vorwissen zu berücksichtigen, auf eine größere Zahl an Nebenbedingungen führt, reduziert sich die Zahl an Parametern deutlich. Betrachtet man beispielsweise ein System mit vier Zuständen und zwei Eingängen, wobei jede Variable zwei linguistische Werte erhält, so führt das Ausschließen eines Eingangs und eines Zustands zu einer Reduktion der Parameter von 64 auf 16. Ähnliches gilt im Übrigen auch, wenn die

Regelbasis vollständig definiert ist (Optimierungsproblem 5.2). In diesem Fall treten die Beschränkungen durch die Zuordnung der Kernpositionsableitungen zu Mengen gemäß Definition 5.1 auf. Die Generalisierungsfähigkeit des RFS kann also durch Einbringen verschiedener Formen von Vorwissen deutlich erhöht werden.

Abschließend sei darauf hingewiesen, dass sich die verschiedenen Formen von Vorwissen auch kombinieren lassen. Ist beispielsweise eine Regelbasis und zusätzlich ein linearer Zusammenhang in einem Hyperquader bekannt, so können die Optimierungsprobleme 5.2 und 5.4 entsprechend kombiniert werden, sodass sowohl Gleichungs- als auch Ungleichungsnebenbedingungen berücksichtigt werden.

5.2.2 Optimierung der Kernpositionen

Will man neben den Kernpositionsableitungen $\mathbf{s}_{\mathbf{w}(\mathbf{j}, \mathbf{q})}^{\mathbf{x}}$, d.h. den Ausgangszugehörigkeitsfunktionen, auch die Lage der Kernpositionen $\mathbf{s}_{\mathbf{j}}^{\mathbf{x}}$, $\mathbf{s}_{\mathbf{q}}^{\mathbf{u}}$, d.h. die Eingangszugehörigkeitsfunktionen, optimieren, entsteht ein multimodales Optimierungsproblem, zu dessen Lösung verschiedene Suchverfahren [142] existieren. Eine Gruppe sind gradientenbasierte Verfahren [142], bei denen sich die Parameter

$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{s}_{\mathbf{j}}^{\mathbf{x}} \\ \mathbf{s}_{\mathbf{q}}^{\mathbf{u}} \end{bmatrix}, \quad \boldsymbol{\xi} \in \mathbb{R}^{l \times 1}, \quad l = \sum_{i=1}^n r_x(i) + \sum_{p=1}^m r_u(p)$$

im aktuellen Iterationsschritt κ aus den Parametern im vorangegangenen Iterationsschritt $\kappa - 1$ und einem Korrekturterm gemäß

$$\boldsymbol{\xi}(\kappa) = \boldsymbol{\xi}(\kappa - 1) - \eta(\kappa - 1) \mathbf{G}(\kappa - 1) \quad (5.15)$$

berechnen. Hierbei gibt η die Schrittweite und \mathbf{G} den Gradienten der Gütefunktion an.

Als Gütefunktion dient wieder der quadratische Fehler zwischen gemessenem \dot{x}_i und dem anhand von $\mathbf{x}^{(v)}$, $\mathbf{u}^{(v)}$ berechneten Zustandswert \hat{x}_i :

$$J_i(\boldsymbol{\xi}) = \frac{1}{2} \sum_{v=1}^N \left(e_i^{(v)} \right)^2 = \frac{1}{2} \sum_{v=1}^N \left(\dot{x}_i^{(v)} - \hat{x}_i^{(v)} \right)^2.$$

Der Index i weist darauf hin, dass bei KRFS insgesamt n Gütefunktionen J_i auftreten, die jeweils für eine Komponente der Zustandsableitung \dot{x}_i gelten. Entsprechend handelt es sich bei der Optimierung der Kernpositionen

grundsätzlich um ein Mehrzieloptimierungsproblem, das zusammenfassend in der Vektorform

$$\min_{\xi} \mathbf{J}(\xi) = \begin{bmatrix} J_1(\xi) \\ \vdots \\ J_n(\xi) \end{bmatrix} \quad (5.16)$$

geschrieben werden kann. Es sei darauf hingewiesen, dass im Gegensatz zur Optimierung der Kernpositionen die Optimierung der Kernpositionsableitungen (Abschnitt 5.2.1) kein Mehrzieloptimierungsproblem darstellt, da die Gütefunktionen bezüglich der Kernpositionsableitungen entkoppelt sind.

Lösung des Mehrzieloptimierungsproblems

Zur Lösung von Mehrzieloptimierungsproblemen existieren verschiedene Ansätze [120]. Das Mehrzieloptimierungsproblem (5.16) ist etwas einfacher zu handhaben als allgemeine Mehrzieloptimierungsprobleme, da alle Kriterien die gleiche Form aufweisen.

Deshalb bietet sich die Anwendung der Gewichtungsmethode an [212]. Diese Methode gewichtet die einzelnen Optimierungskriterien in geeigneter Weise, wodurch das Mehrzieloptimierungsproblem (5.16) in ein einzelnes Optimierungsproblem der Form

$$\min_{\xi} J(\xi) = \begin{bmatrix} \tau_1 & \cdots & \tau_n \end{bmatrix} \begin{bmatrix} J_1(\xi) \\ \vdots \\ J_n(\xi) \end{bmatrix} \quad (5.17)$$

überführt wird. Entscheidend für die erfolgreiche Anwendung ist die Definition eines geeigneten Vektors $\boldsymbol{\tau}^T$, der die einzelnen Optimierungskriterien ausgewogen gewichtet. Würde man die jeweiligen quadratischen Fehler gleich gewichten, so würden die Zustandsgrößen mit dem größten Wertebereich den Optimierungsprozess dominieren. Um dies auszuschließen, werden die quadratischen Fehler gemäß dem Verhältnis der Varianzen der jeweiligen gemessenen Zustandsableitungen \dot{x}_i gewichtet und damit normiert. Bezeichnet man mit $\sigma_{\dot{x}_i}$ die Varianz der Messdaten der i-ten Zustandsableitung, so können die Einträge des Gewichtungsvektors durch

$$\tau_i = \frac{1}{n-1} \cdot \frac{\sum_{j=1, j \neq i}^n \sigma_{\dot{x}_j}}{\sum_{j=1}^n \sigma_{\dot{x}_j}} \quad (5.18)$$

definiert werden.

Bestimmung der Gradienten

Der nächste Schritt ist die Bestimmung des in (5.15) benötigten Gradienten der Gütefunktion (5.17) zu

$$\mathbf{G} = \frac{\partial J(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = \frac{\partial (\boldsymbol{\tau}^T \cdot \mathbf{J}(\boldsymbol{\xi}))}{\partial \boldsymbol{\xi}} = \boldsymbol{\tau}^T \cdot \frac{\partial \mathbf{J}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}},$$

wobei

$$\frac{\partial \mathbf{J}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = \begin{pmatrix} \frac{\partial J_1(\boldsymbol{\xi})}{\partial \xi_1} & \dots & \frac{\partial J_1(\boldsymbol{\xi})}{\partial \xi_l} \\ \vdots & \ddots & \vdots \\ \frac{\partial J_n(\boldsymbol{\xi})}{\partial \xi_1} & \dots & \frac{\partial J_n(\boldsymbol{\xi})}{\partial \xi_l} \end{pmatrix} \quad (5.19)$$

die Jacobi-Matrix bezüglich des Parametervektors ist.

Für die jeweiligen Einträge von (5.19), also die Ableitung einer Gütefunktion J_i , gilt allgemein

$$\frac{\partial J_i(\boldsymbol{\xi})}{\partial \xi_k} = - \sum_{v=1}^N \left(\dot{x}_i^{(v)} - \hat{x}_i^{(v)} \right) \cdot \frac{\partial \hat{x}_i^{(v)}(\boldsymbol{\xi})}{\partial \xi_k}$$

mit

$$\frac{\partial \hat{x}_i^{(v)}(\boldsymbol{\xi})}{\partial \xi_k} = \sum_{\mathbf{j}, \mathbf{q}} s_{w_i(\mathbf{j}, \mathbf{q})}^{\dot{x}_i} \cdot \frac{\partial \Xi(\mathbf{j}, \mathbf{q})(\mathbf{x}^{(v)}, \mathbf{u}^{(v)})}{\partial \xi_k}.$$

Die Ableitung von $\Xi(\mathbf{x}^{(v)}, \mathbf{u}^{(v)})$ nach ξ_k , d.h. der Kernposition $s_{r_k}^{x_k}$ bzw. $s_{q_p}^{x_p}$, berechnet sich beispielsweise zu

$$\frac{\partial \Xi(\mathbf{x}^{(v)}, \mathbf{u}^{(v)})}{\partial s_{r_k}^{x_k}} = \frac{\partial \mu_{j_k}^{x_k}(x_k^{(v)})}{\partial s_{r_k}^{x_k}} \prod_{i=1, i \neq k}^n \mu_{j_i}^{x_i}(x_i^{(v)}) \prod_{p=1}^m \mu_{q_p}^{u_p}(u_p^{(v)}), \quad (5.20)$$

da in jeder Regel nur eine Zugehörigkeitsfunktion pro Dimension aktiviert wird, sodass der Parameter $s_{r_k}^{x_k}$ nur einmal auftritt.

Schließlich gilt für die Ableitung der dreieck- und rampenförmigen Zugehörigkeitsfunktionen (3.3):

$$\frac{\partial \mu_{j_k}^{x_k}(x_k^{(v)})}{\partial s_{r_k}^{x_k}} = \begin{cases} \frac{(x_k^{(v)} - s_{j_i}^{x_i})}{(s_{j_i}^{x_i} - s_{j_i-1}^{x_i})^2} \cdot \delta(s_{j_i-1}^{x_i}, s_{j_i}^{x_i}, x_k^{(v)}) \\ \text{wenn } r_k = j_i - 1, \\ \\ -\frac{(x_k^{(v)} - s_{j_i-1}^{x_i})}{(s_{j_i}^{x_i} - s_{j_i-1}^{x_i})^2} \cdot \delta(s_{j_i-1}^{x_i}, s_{j_i}^{x_i}, x_k^{(v)}) \\ -\frac{(x_k^{(v)} - s_{j_i+1}^{x_i})}{(s_{j_i+1}^{x_i} - s_{j_i}^{x_i})^2} \cdot \delta(s_{j_i}^{x_i}, s_{j_i+1}^{x_i}, x_k^{(v)}) \\ \text{wenn } r_k = j_i, \\ \\ \frac{(x_k^{(v)} - s_{j_i}^{x_i})}{(s_{j_i+1}^{x_i} - s_{j_i}^{x_i})^2} \cdot \delta(s_{j_i}^{x_i}, s_{j_i+1}^{x_i}, x_k^{(v)}) \\ \text{wenn } r_k = j_i + 1, \\ \\ 0 \text{ sonst,} \end{cases} \quad (5.21)$$

wobei

$$\delta(a, b, x_k^{(v)}) = \begin{cases} 1 & \text{wenn } a < x_k^{(v)} \leq b, \\ 0 & \text{sonst.} \end{cases}$$

Die Ableitungen nach den Kernpositionen s_q^u ergeben entsprechende Ergebnisse.

Genau genommen ist die Ableitung der Zugehörigkeitsfunktion in der Kernposition nicht differenzierbar. Vereinfachend kann man die Ableitung in der Kernposition entweder zu null oder auf den links- oder rechtsseitigen Grenzwert setzen. Diese Vereinfachung ist ohne praktische Einschränkung möglich, da dies nur in dem unwahrscheinlichen Fall auftritt, wenn ein Datenpaar exakt auf der Kernposition liegt. Setzt man die Ableitung zu null, bedeutet dies lediglich, dass das Datenpaar in diesem Iterationsschritt nicht betrachtet wird.

5.2.3 Gleichzeitige Optimierung aller Parameter

Nachdem in den vorigen Abschnitten die Lernverfahren für die Kernpositionen und die Kernpositionsableitungen vorgestellt wurden, werden diese, ähnlich wie in [84], nun in einen Gesamtalgorithmus integriert, sodass eine gleichzeitige Optimierung aller Parameter erfolgen kann:

Algorithmus 5.1.

1. Initialisierungsschritt: Die Anzahl der Zugehörigkeitsfunktionen und die Lage der Kernpositionen in jeder Dimension wird festgelegt. Außerdem werden die Kernpositionsableitungen mittels Least Squares oder QP initialisiert.
2. Lernen der Kernpositionen: Aktualisieren der Kernpositionen ξ durch Anwendung des Gradientenverfahrens (Abschnitt 5.2.2)
3. Lernen der Kernpositionsableitungen: Aktualisieren der Kernpositionsableitungen $s^{\dot{x}_i}$ durch Anwendung des Least-Squares-Verfahrens oder der QP (Abschnitt 5.2.1).
4. Überprüfung des Abbruchkriteriums: Unterschreitet die Parameteränderung eine vorgegebene Grenze oder steigt der Validierungsfehler an, wird der Algorithmus beendet. Ansonsten gehe zu Schritt 2.

Der Algorithmus bietet den Vorteil, unabhängig vom Grad des Vorwissens anwendbar zu sein. Ist beispielsweise eine komplette Regelbasis oder Vorwissen über das System in Bereichen des Zustandsraums gegeben, so werden die Kernpositionsableitungen mit Beschränkungen optimiert. Ist kein Vorwissen vorhanden, sondern nur die Zahl der Kernpositionen in jeder Dimension festgelegt, wird entsprechend ohne Beschränkungen optimiert.

Einschrittprädiktion und Simulation

Einen wichtigen Aspekt bei der Modellierung dynamischer Systeme stellt der bei der Optimierung verwendete Prädiktionsfehler dar, der über das Gütekriterium 5.6 definiert wird. Dieser setzt sich in der Parameteroptimierung aus Algorithmus 5.1 aus den Differenzen zwischen gemessenen Werten und den mithilfe *aktueller Messdaten* geschätzten Modellausgängen zusammen. Durch die Verwendung aktueller Messdaten ergibt sich ein *einschrittoptimales* Modell. Alternativ kann man die geschätzten Modellausgänge aber auch mithilfe der *in den vorigen Zeitschritten geschätzten Daten* berechnen [142]. So optimierte Modelle sind dann *mehrschrittoptimal*, so dass sie im Simulationsbetrieb zumeist genauer sind. Andererseits führt die Optimierung mehrschrittoptimaler Modelle auf ein Optimierungsproblem, das bezüglich der Kernpositionsableitungen nicht mehr konvex ist. Man muss dann auch zur Optimierung der Kernpositionsableitungen ein Gradientenverfahren einsetzen, was aufgrund des deutlich größeren Suchraums zur Konvergenz in lokale Minima führen kann und die Rechenzeit deutlich erhöht.

Als Kompromiss wird in dieser Arbeit, wie u.a. in [142] vorgeschlagen, der Validierungsfehler aus dem simulierten Modell gebildet. Dann bleibt das Optimierungsproblem bezüglich der Kernpositionsableitungen konvex, im Abbruchkriterium wird allerdings die Performance der Mehrschrittprädiktion und nicht die Einzschrittprädiktion bewertet. Die so erhaltenen Modelle haben, wie beispielsweise das zur Fehlerdetektion verwendete Dreitank-System (Kapitel 7), auch im Simulationsbetrieb eine sehr gute Performance gezeigt.

5.3 Strukturoptimierung

In den vorangegangenen Abschnitten wurde gezeigt, wie die Parameter von KRFS optimiert werden, sodass sie nichtlineare Systeme nachbilden können. Hierbei wurde angenommen, dass die Struktur, d.h. die Zahl an Zugehörigkeitsfunktionen für jede Eingangs- und Zustandsvariable, bekannt ist. Hierzu sind allerdings qualitative Informationen über die Nichtlinearitäten im Prozess notwendig. Diese können allerdings in vielen realen Systemen nicht als gegeben vorausgesetzt werden, entweder weil kein Expertenwissen vorhanden ist oder weil sich das System nur schlecht durch Menschen analysieren lässt. In solchen Fällen kann die Struktur des KRFS auch ausschließlich auf Basis von Messdaten bestimmt werden.

Wir beschäftigen uns also nun mit Teilproblem 2 von Modellierungsproblem 5.1. Wie schon in Kapitel 2.2.2 angesprochen, existieren in der Literatur eine Vielzahl von Verfahren zur Strukturidentifikation von Fuzzy-Systemen, die sich grob in iterativ-heuristische Verfahren, Metaheuristiken und Clusteringverfahren einteilen lassen. Bei der konkreten Anwendung auf rekurrente Fuzzy-Systeme haben Clusteringverfahren allerdings den Nachteil, dass sie eine Streupartitionierung mit mehrdimensionalen Zugehörigkeitsfunktionen erzeugen. Da RFS prinzipiell mit eindimensionalen Zugehörigkeitsfunktionen arbeiten und eine Gitterpartitionierung verwenden, bietet sich eine Anwendung von Clusteringverfahren nicht an. Metaheuristiken, wie genetische Programmierung, dagegen lassen sich anwenden, wie in [196] gezeigt wurde. Allerdings ergeben sich vergleichsweise hohe Rechenzeiten. Außerdem gestaltet sich die Einstellung geeigneter Parameter der genetischen Programmierung als schwierig.

Deshalb wird im Folgenden ein Baumstruktur-Algorithmus für KRFS vorgestellt, der in die Gruppe der iterativ-heuristischen Verfahren eingeordnet werden kann. Solch ein Algorithmus wurde in [142] für radiale Basisfunktionennetze, die TSK Fuzzy-Systemen sehr ähnlich sind, vorgestellt.

Er erzeugt zwar eine achsenorthogonale, aber keine gitterähnliche Partitionierung des Eingangsraums. Dagegen bildet der Strukturalgorithmus in [27] eine gitterähnliche Partitionierung für TSK Fuzzy-Systeme. Allerdings hat dieser Algorithmus den Nachteil, dass er Modelle mit einer sehr großen Anzahl von Regeln generiert, wodurch die entstehenden Modelle schwieriger zu interpretieren sind.

Im Folgenden soll nun ein Baumstruktur-Algorithmus zur Strukturoptimierung von KRFS vorgestellt werden, der sich insbesondere zur Bildung von regelbasierten Systemen eignet, deren Dynamik linguistisch interpretierbar ist. Hierbei wird nicht nur die Eingangspartitionierung, sondern eine vollständige Regelbasis, die die Dynamik des Systems beschreibt, identifiziert. Besondere Bedeutung wird hierbei der Interpretierbarkeit des resultierenden KRFS beigemessen.

5.3.1 Partitionierung des Eingangsraums

Der nun vorgestellte Algorithmus konstruiert schrittweise eine vollständige Partitionierung des Eingangsraums, d.h., die Zahl an Zugehörigkeitsfunktionen (Kernpositionen) in jeder Dimension wird auf Basis des Trainingsdatensatzes bestimmt.

Algorithmus 5.2.

1. Starte mit einer Zugehörigkeitsfunktion für jeden Eingang und Zustand, initialisiere die Kernpositionsableitung mittels des Optimierungsverfahrens aus Abschnitt 5.2.1 und bestimme den mittleren quadratischen Fehler MSE.
2. Führe die folgenden Schritte für jede Dimension $l = 1, \dots, n + m$ des Eingangsraums aus:
Falls die betrachtete l -te Variable nur eine Zugehörigkeitsfunktion zugeordnet hat:
 - 2a) Setze zwei Kernpositionen auf den minimalen und maximalen Wert des Trainingsdatensatzes.
 - 2b) Optimierte die Kernpositionsableitungen des um diese Kernpositionen erweiterten RFS und berechne den MSE.

ansonsten:

- 2c) Ermittle den Bereich zwischen zwei aufeinander folgenden Kernpositionen der l -ten Dimension, in dem der lokale mittlere quadratische Fehler am größten ist. Abbildung 5.4 zeigt Beispiele

für einen zwei-dimensionalen Eingangsraum. Die Bereiche für jede Dimension mit größtem MSE sind grau markiert.

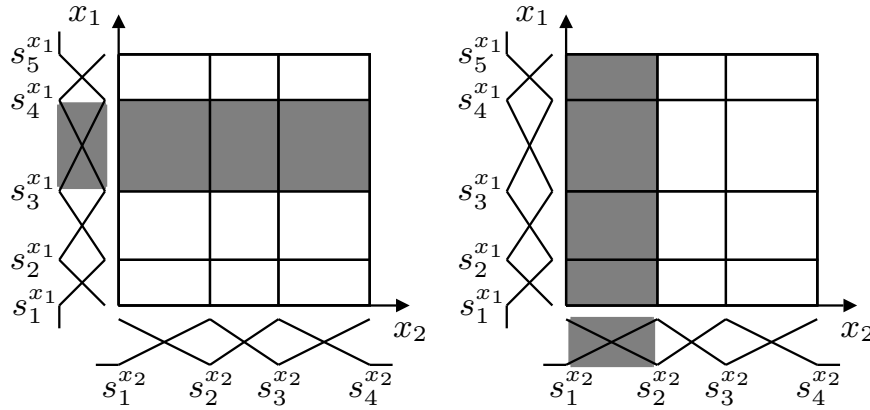


Abbildung 5.4: Beispiele für Bereiche mit größtem Fehler in x_1 (links) und x_2 (rechts)

- 2d) Berechne den Fehlerschwerpunkt für den ermittelten Bereich gemäß

$$x_{r,\text{fsp}} = \frac{\sum_{v|x_l^{(v)} \in X_{l,\text{max}}} x_l^{(v)} \cdot e_{\text{ges}}^{(v)}}{\sum_{v|x_l^{(v)} \in X_{l,\text{max}}} e_{\text{ges}}^{(v)}},$$

wobei $X_{l,\text{max}}$ den Bereich mit größtem MSE in der l -ten Dimension bezeichnet und

$$e_{\text{ges}}^{(v)} = \boldsymbol{\tau}^T \cdot \begin{pmatrix} (e_1^2)^{(v)} \\ \vdots \\ (e_n^2)^{(v)} \end{pmatrix}$$

den gewichteten quadratischen Fehler aller Komponenten des Zustandsvektors bezeichnet. Eine neue Kernposition wird nun vorübergehend am Fehlerschwerpunkt hinzugefügt, d.h., $s_{ji,\text{neu}}^{x_i} = x_{l,\text{fsp}}$.

- 2e) Optimierte die Kernpositionsableitungen des RFS mit der neu (und vorübergehend) hinzugefügten Kernposition $s_{ji,\text{neu}}^{x_i}$ mittels Least-Squares-Optimierung und berechne den gesamten MSE.

Mit den vorigen Schritten 2a) - 2b) bzw. 2c) - 2e) wurden zwei bzw. eine neue Kernposition für *jede* Eingangsdimension berechnet.

3. Ermittle die Dimension l^* und damit die Kernposition(en), die den kleinsten MSE ergeben und ergänze sie zum RFS.

4. (*optional*) Falls der MSE unter eine vorher festgelegte Grenze $\alpha^* = \alpha_{gr} \text{MSE}_0$ fällt, wobei $\alpha_{gr} = 0.05, \dots, 0.2$, optimiere die Kernpositionen mit der in Abschnitt 5.2.3 beschriebenen Parameteroptimierung.
5. Falls ein Abbruchkriterium erfüllt ist, beende die Strukturoptimierung, ansonsten gehe zu Schritt 3.

Abbildung 5.5 illustriert den Ablauf des Algorithmus für einen zweidimensionalen Eingangsraum. Wie bereits angesprochen, liegt der große Vorteil des RFS in der guten Interpretierbarkeit. Wird allerdings bei der Modellierung komplexer nichtlinearer Systeme nur die Approximationsgüte des RFS betrachtet, entstehen Modelle mit sehr vielen Zugehörigkeitsfunktionen in jeder Dimension, was zu einer exponentiell steigenden Zahl an Regeln führt. Diese Modelle sind kaum mehr interpretierbar. Es muss also ein geeignetes Abbruchkriterium definiert werden, das bei zufriedenstellender Approximationsgüte die Interpretierbarkeit erhält.

Dafür wird das folgende Eingangsstrukturkriterium (ESK), das Akaike's Informationskriterium [9] ähnelt, wie folgt definiert:

$$\text{ESK}_\kappa = \ln(\boldsymbol{\tau}^T \cdot \mathbf{MSE}_\kappa) + \rho_{\text{ESK}} \cdot R_\kappa,$$

wobei \mathbf{MSE}_κ der Vektor des MSE in jeder Zustandsvariablen ist, $\boldsymbol{\tau}^T$ ist der Gewichtungsvektor gemäß Gl. (5.18), R_κ ist die Zahl an Regeln im κ -ten Iterationsschritt und ρ_{ESK} ist ein Parameter, der die Gewichtung von Modellgüte und Interpretierbarkeit einstellt. Je höher ρ_{ESK} gewählt wird, desto niedriger ist die Zahl an Regeln nach der Strukturoptimierung. Eine sinnvolle Wahl ist es, ρ_{ESK} derart zu normieren, dass beide Anteile des ESK gleiche Wertebereiche annehmen. Dies kann z.B. durch $\rho_{\text{ESK}} = \ln(\beta^* \cdot \boldsymbol{\tau}^T \cdot \text{MSE}_0) / R_w$ erfolgen, wobei R_w die gewünschte Zahl an Regeln nach der Strukturoptimierung und β^* die gewünschte Verbesserung der Approximationsgüte bezogen auf den initialen MSE ist. Falls nach einer Iteration das ESK ansteigt, wird die Optimierung abgebrochen. Zusätzlich zum ESK wird auch bei diesem Algorithmus mittels eines Validierungsdatensatzes die Generalisierungsfähigkeit des RFS überwacht, um eine Überanpassung an den Datensatz zu vermeiden. Falls der Validierungsfehler ansteigt, wird die Optimierung beendet.

Schritt 5 der Strukturoptimierung ist optional. Im Allgemeinen ist zu erwarten, dass die hinzugefügte Kernposition nicht das Optimum trifft, sondern nur eine gute Näherung ist. Falls zusätzlich eine Parameteroptimierung durchgeführt wird, kann der MSE weiter reduziert und ein (lokales) Optimum gefunden werden. Diese Reduktion des MSE kann dazu

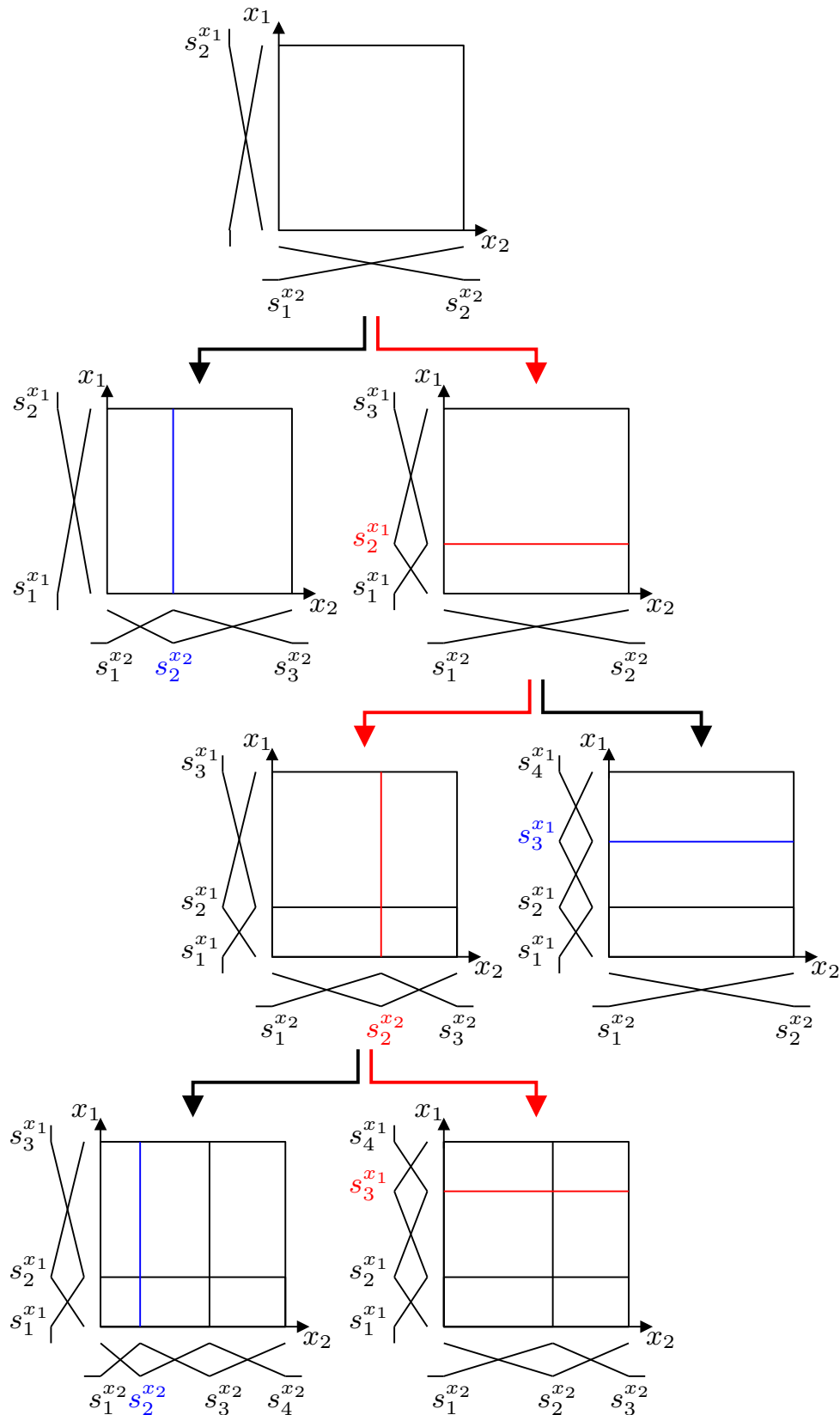


Abbildung 5.5: Ablauf der Strukturoptimierung für RFS für einen zwei-dimensionalen Eingangsraum. Die Optimierung erfolgt entlang des roten Pfeils. In Richtung des schwarzen Pfeils wird nicht weiter optimiert.

führen, dass das ESK bereits früher ansteigt und damit die Optimierung beendet. Da jede zusätzlich eingefügte Kernposition die Zahl an Regeln deutlich erhöht, kann ein früherer Abbruch der Optimierung zu einer geringeren Zahl an Regeln bei ähnlicher Modellgüte führen und damit die Interpretierbarkeit verbessern. Allerdings wird auch die Rechenzeit wegen der nichtlinearen Optimierung ansteigen. Als sinnvoller Kompromiss hat sich herausgestellt, die Parameteroptimierung in den ersten Schritten nicht durchzuführen. Erst wenn die Struktur des RFS so weit aufgebaut wurde, dass das System ausreichend gut modelliert werden kann, wird die Parameteroptimierung angewendet.

Algorithmus 5.2 hat noch eine zusätzliche Eigenschaft. Da man mit einer Zugehörigkeitsfunktion pro Dimension startet, kann man nach der Optimierung feststellen, welche Zustände und Eingänge für die Modellierung des Systems tatsächlich relevant sind. Wird im Laufe der Optimierung einer Eingangs- oder Zustandsgröße keine weitere Zugehörigkeitsfunktion zugeordnet, so hat diese Variable einen konstanten Zugehörigkeitswert von eins, d.h., die Variable beeinflusst das Verhalten des RFS nicht. Entsprechend ist diese Variable auch nicht relevant und kann für die weitere Modellbildung vernachlässigt werden.

5.3.2 Reduktion der Kernpositionsableitungen

Der im vorangegangenen Abschnitt beschriebene Algorithmus optimiert die Struktur des Eingangsraums des RFS. Das bedeutet, dass den Eingangs-, und Zustandsgrößen eine entsprechende Zahl an Kernpositionen zuordnet wird. Um nun eine vollständige Regelbasis zu identifizieren, muss auch die Struktur des Ausgangsraums, also die Zahl an Kernpositionsableitungen pro Zustandsableitung identifiziert werden. Dies ist, wie bereits angesprochen, auch deshalb wünschenswert, weil die linguistische Interpretierbarkeit des RFS bei nicht eingeschränkter Zahl der Kernpositionsableitungen nicht mehr gegeben ist.

Der folgende Algorithmus reduziert nun auf Basis des zuvor gelernten RFS die Zahl der Kernpositionsableitungen auf eine linguistisch interpretierbare Anzahl. Dies kann für jede Zustandsableitung gesondert erfolgen, da die Optimierungsprobleme 5.2 entkoppelt sind.

Algorithmus 5.3.

1. *Hierarchisches Clustern*: Fasse n Kernpositionsableitungen mit ähnlichen Wertebereichen mittels eines hierarchischen Clusteralgorithmus [137] zusammen. Hierbei wird nur die qualitative Information

des Cluster-Algorithmus genutzt. Das heißt, dass nur die Information, welche Kernpositionsableitungen zusammengefasst werden, weiterverwendet wird. Die zusammengefassten Kernpositionsableitungen werden also nicht durch ihren Mittelwert o.Ä. ersetzt.

2. *Optimierung bei vollständiger Regelbasis:* Optimierte die reduzierte Zahl an Kernpositionsableitungen mithilfe von Optimierungsproblem 5.2. Berechne den entstehenden mittleren quadratischen Fehler MSE.
3. *Überprüfen des Abbruchkriteriums:* Falls das weiter unten beschriebene Abbruchkriterium (5.22) erfüllt ist, beende den Algorithmus, ansonsten erhöhe die Zahl der zusammenzufassenden Kernpositionsableitungen und gehe zu Schritt 1.

Wie bereits bei der Optimierung der Eingangsstruktur, muss auch in diesem Fall ein Kompromiss zwischen Modellierungsgenauigkeit und linguistischer Interpretierbarkeit gefunden werden. Hierzu wird das Ausgangsstrukturkriterium ASK

$$\text{ASK}_\gamma = \alpha_{\text{ASK}} \cdot \text{MSE}_\gamma + e^{0.2(\gamma - \gamma_0)} \quad (5.22)$$

definiert, wobei γ die Zahl an Clustern und MSE_γ den MSE bezeichnet, der sich bei Optimierung mit γ Clustern ergibt. α_{ASK} und γ_0 sind einstellbare Parameter, die wie folgt interpretiert werden können. Mit γ_0 kann die Zahl an Kernpositionsableitungen, für die die linguistische Interpretierbarkeit erhalten bleibt, typischerweise $\gamma_0 \in [7, 9]$, eingestellt werden. Für den Gewichtungsfaktor der Modellgüte α_{ASK} ist eine Normierung auf den ursprünglichen MSE ohne Zusammenfassen $\text{MSE}_{\max(\gamma)}$ gemäß

$$\alpha_{\text{ASK}} = \frac{1}{\alpha_{\text{ASK}}^* \cdot \text{MSE}_{\max(\gamma)}}$$

sinnvoll. Durch α_{ASK}^* kann dann prozentual angegeben werden, wieviel Modellgüte man für eine bessere Interpretierbarkeit opfern möchte.

5.3.3 Reduktion der Regeln

Wie bereits in Abschnitt 5.2.1 angesprochen, ist nicht nur die Reduktion an Kernpositionsableitungen, sondern auch die Reduktion der Regeln wünschenswert, da dies zu einer flexibleren Partitionierung des Eingangsraums führt. Der nun vorgestellte Algorithmus reduziert schrittweise die Regeln,

die bereits durch andere Regeln weitestgehend abgedeckt werden und somit annähernd redundant sind.

Algorithmus 5.4.

1. *Initialisierung*: Optimierte das KRFS gemäß Optimierungsproblem 5.1 und berechne den initialen mittleren quadratischen Fehler des Validierungsdatensatzes $\text{MSE}_{\text{init}} = \text{MSE}_{\text{valid}}^0$.
2. *Berechne Distanzmaß*: Berechne für alle Regeln der Regelbasis das Distanzmaß d_w

$$d_w = \frac{\|\mathbf{s}_{\mathbf{w},\text{akt}}^{\dot{\mathbf{x}}} - \mathbf{s}_{\mathbf{w},\text{interp}}^{\dot{\mathbf{x}}}\|_2}{\|\mathbf{s}_{\mathbf{w},\text{akt}}^{\dot{\mathbf{x}}}\|_2}, \quad w = 1, \dots, R$$

zwischen dem aktuellen Wert der Kernpositionsableitung $\mathbf{s}_{\mathbf{w},\text{akt}}^{\dot{\mathbf{x}}}$ und dem sich gemäß Gl. (5.11) aus der Interpolation der benachbarten Kernpositionsableitungen ergebenden Wert $\mathbf{s}_{\mathbf{w},\text{interp}}^{\dot{\mathbf{x}}}$. Falls der niedrigste Wert aller Distanzmaße eine vorgegebene Grenze überschreitet, d.h. $\min d_w > \delta_0$, beende den Algorithmus, ansonsten wähle Kernpositionsableitung mit niedrigstem Distanzmaß und gehe zu Schritt 3.

3. *Optimierung mit reduzierten Regeln*: Definiere Nebenbedingungen der Form (5.11), sodass die ausgewählte Kernpositionsableitung redundant wird. Optimierte die Kernpositionsableitungen mit diesen Nebenbedingungen gemäß Optimierungsproblem 5.3 und berechne den entstehenden mittleren quadratischen Fehler $\text{MSE}_{\text{valid}}^{\text{neu}}$.
4. *Überprüfen des Abbruchkriteriums*: Falls der MSE des Validierungsdatensatzes stärker ansteigt als eine vorgegebene Grenze, d.h. $\text{MSE}_{\text{valid}}^{\text{neu}} - \text{MSE}_{\text{valid}}^0 > \delta_1$, beende den Algorithmus, ansonsten entferne die Regel aus der Regelbasis, aktualisiere $\text{MSE}_{\text{valid}}^0 \leftarrow \text{MSE}_{\text{valid}}^{\text{neu}}$ und gehe zu Schritt 1.

In Algorithmus 5.4 sind noch die Parameter δ_0 und δ_1 einzustellen. Der Parameter δ_0 gibt eine untere Grenze an, ab der sich eine Kernposition deutlich von den interpolierten benachbarten Kernpositionsableitungen unterscheidet. Je größer δ_0 gewählt wird, desto mehr Regeln werden auf Relevanz untersucht. Die Entscheidung, ob eine Regel redundant ist, ist dann hauptsächlich von der Wahl von δ_1 abhängig. Je größer δ_1 gewählt wird, desto mehr Regeln werden entfernt bei gleichzeitiger Inkaufnahme eines höheren MSE. Sinnvoll ist eine Wahl von $\delta_1 = \delta^* \text{MSE}_{\text{init}}$, sodass δ^* die

maximal erlaubte prozentuale Erhöhung des initialen MSE ist. Es sei angemerkt, dass auch die Definition eines Strukturkriteriums ähnlich dem ASK möglich ist.

5.4 Auswahl der relevanten Eingangs- und Zustandsgrößen

Entscheidend für eine gute Modellbildung ist die Wahl der relevanten Größen, die sich im Falle rekurrenter Fuzzy-Systeme aus den Eingangs- und Zustandsgrößen zusammensetzen. Wird nur ein Teil dieser Größen zur Modellbildung verwendet, so kann das dynamische Verhalten des Systems nicht adäquat nachgebildet werden. Werden dagegen nicht relevante Größen mit in den Modellbildungsprozess einbezogen, so sind mehr Parameter als notwendig zu optimieren, was zur Überanpassung an den Datensatz führt.

Die in den vorangegangenen Abschnitten beschriebenen Algorithmen enthalten bereits die Möglichkeit, relevante Größen zu ermitteln. Ist ein Eingang bzw. Zustand nicht relevant, so werden diesem keine weiteren Kernpositionen zugeordnet. Größen, die nur eine Kernposition aufweisen, haben also keinen Einfluß auf die Systemdynamik und können entfernt werden. Dies gilt allerdings nur bezogen auf das Gesamtsystem, es werden also nur Größen aussortiert, die das System überhaupt nicht beeinflussen. Dagegen lässt sich die Frage, welche Zustandsableitung direkt von welchen Eingängen bzw. Zuständen beeinflusst wird, nicht klären.

Bei RFS in ihrer Grundform (3.5) wird angenommen, dass die Zustandsableitungen \dot{x}_i direkt von allen Zustands- und Eingangsvariablen, die in den Mengen \mathcal{X} und \mathcal{U} zusammengefasst werden, abhängen. Man nimmt also $\dot{x}_i = f_i(\mathbf{x}, \mathbf{u})$ an. In nahezu allen realen Systemen gilt allerdings, dass die Zustandsableitung \dot{x}_i nur von einer Teilmenge der Zustands- und Eingangsvariablen \mathcal{Z}_s abhängig ist. Es gilt also tatsächlich $\dot{x}_i = f_i(\mathbf{x}_s, \mathbf{u}_s)$, wobei $(\mathbf{x}_s, \mathbf{u}_s) \in \mathcal{Z}_s \subseteq \mathcal{Z} = \{x_1, \dots, x_n, u_1, \dots, u_m\}$ gilt. Zur Modellbildung ist es also wünschenswert nur die tatsächlich relevanten Zustände und Eingänge zu berücksichtigen. Außerdem kann durch solche Einschränkungen, wie bereits in Abschnitt 5.2.1 besprochen, die Zahl an Optimierungsparametern und damit eine Überanpassung an die Datensätze verringert werden [172].

Wir widmen uns also nun der Lösung von Teilproblem 1 von Modellierungsproblem 5.1, das wie folgt konkretisiert wird:

Modellierungsproblem 5.2. Gegeben seien Datensätzen (5.1) bzw. (5.2) des Systems und eine Partitionierung des Eingangsraums (Zahl an Zugehö-

rigkeitsfunktionen für jeden Zustand und Eingang) des KRFS. Bestimme für jede Zustandsableitung \dot{x}_i die Teilmenge \mathcal{Z}_s von relevanten Zustands- und Eingangsvariablen.

Grundsätzlich handelt es sich bei Problem 5.2 um ein Problem der Teilmengenauswahl (engl. Subset Selection). Hierzu gibt es verschiedene Ansätze, die sich grob in Verfahren mit orthogonaler Projektion, d.h. die schon erwähnten orthogonalen Least-Squares-Verfahren, sowie Auswahlverfahren aufteilen lassen [136, 142]. Die Anwendung des OLS-Verfahrens sowie der Vorwärts- und Rückwärtsauswahl zur Variablenauswahl wurde in [169] untersucht und verglichen. Das OLS-Verfahren arbeitet direkt mit den multilineareren Polynomen. Aufgrund der Stetigkeit des RFS entstehen allerdings Zwangsbedingungen zwischen den Koeffizienten verschiedener multilinearer Polynome, die im OLS-Verfahren nicht berücksichtigt werden. Entsprechend ist die Variablenauswahl nur lokal, also für jeden Hyperquader einzeln, gültig. Diese Einschränkung kann bei Auswahlverfahren, wie der Vorwärts- und Rückwärtsauswahl, vermieden werden. Prinzipiell hat die Vorwärtsauswahl dann Vorteile, wenn nur wenige Elemente aus der Menge relevant sind, die Rückwärtsauswahl entsprechend, wenn viele Elemente relevant sind. In der Anwendung auf die Variablenauswahl bei KRFS bietet sich die Rückwärtsauswahl an, insbesondere weil das KRFS zunächst vollständig definiert ist, alle Eingänge und Zustände also berücksichtigt werden.

Die konkrete Lösung erfolgt wiederum über die hybride Systemdarstellung des KRFS. Wie bereits in Abschnitt 5.2.1 angesprochen, können Eingänge und Zustände durch entsprechende Nebenbedingungen ausgeschlossen werden. Startet man nun mit dem kompletten Fuzzy-System, kann gemäß Optimierungsproblem 5.4 jeweils eine Zustands- oder Eingangsgröße durch Beschränkungen ausgeschlossen und der daraus resultierende RMSE des Validierungsdatensatzes beobachtet werden. Ergibt sich nun eine deutliche Erhöhung des RMSE, können wir schlussfolgern, dass die entsprechende Variable relevant ist. Bleibt der Fehler dagegen auch nach dem Ausschluss der Variablen klein, ist die Variable offenbar nicht relevant. In jedem Schritt wird nun die Variable ausgeschlossen, die als nicht relevant klassifiziert wird und die kleinste Erhöhung im RMSE erzeugt. Dies wird solange wiederholt, bis alle verbleibenden Variablen als relevant klassifiziert sind. Diese Vorgehensweise wird in der Literatur zumeist als *Rückwärtsauswahl* bezeichnet [26, 136]. Der Ansatz wird im folgenden Algorithmus zusammengefasst:

Algorithmus 5.5.

Für jede Zustandsableitung \dot{x}_i , $i = 1, \dots, n$:

1. Definiere die aktuelle Teilmenge von relevanten Variablen $\mathcal{Z}_a = \mathcal{Z}$.
2. Optimierte die Kernpositionsableitungen ohne Beschränkungen und berechne den RMSE_0 .
3. Für jede Variable z_j der Teilmenge \mathcal{Z}_a :
 - 3a) Definiere Beschränkungen, sodass die Variable z_j ausgeschlossen wird.
 - 3b) Optimierte die Kernpositionsableitungen mit den Beschränkungen aus Schritt 3.1 und berechne den RMSE_j .
4. Bestimme die Variable z_j^* mit dem kleinsten RMSE. Wenn $\text{RMSE}_{j^*} - \text{RMSE}_0 < \alpha_{\text{EA}}$ gilt, wobei α_{EA} ein gegebener Schwellwert ist, entferne z_j^* aus der Teilmenge, d.h., $\mathcal{Z}_a \leftarrow \mathcal{Z}_a \setminus \{z_j^*\}$, aktualisiere $\text{RMSE}_0 \leftarrow \text{RMSE}_{j^*}$ und gehe zu Schritt 3. Ansonsten, falls $i < n$, gehe zu Schritt 1, sonst beende die Variablenauswahl.

Der Algorithmus 5.5 arbeitet global in dem Sinn, dass die Optimierung über alle Parameter des KRFS durchgeführt wird. Insbesondere für Systeme höherer Ordnung führt dies zu Optimierungsproblemen mit vielen Parametern, die oft schlecht konditioniert sind. Dies lässt sich vermeiden, indem jeder elementare Hyperquader individuell betrachtet wird, d.h., die relevanten Zustände und Eingänge werden für jeden elementaren Hyperquader bestimmt. Die Ergebnisse für die einzelnen Hyperquader werden dann zu einem Gesamtergebnis gemittelt. Neben der Verkleinerung des Optimierungsproblems erlaubt dieser Ansatz auch die Extraktion von lokaler Information für jeden Hyperquader und damit für Bereiche des Zustandsraums. Es ergibt sich die folgende Modifikation von Algorithmus 5.5:

Algorithmus 5.6.

Für jede Zustandsableitung \dot{x}_i , $i = 1, \dots, n$:

1. Führe für jeden elementaren Hyperquader $l = 1, \dots, K$ die Schritte 1.-4. von Algorithmus 5.5 unter Berücksichtigung der Kernpositionsableitungen des l -ten Hyperquaders durch.
2. Berechne den Mittelwert der Ergebnisse für jeden elementaren Hyperquader von Schritt 2, um das globale Ergebnis zu erhalten. Falls der Mittelwert größer als ein Grenzwert β , der normalerweise zu 0.5 gewählt wird, ist, klassifiziere die Variable als relevant.

Der einzige Parameter, der in den Algorithmen 5.5 und 5.6 festgelegt werden muss, ist der Schwellwert α_{EA} . Prinzipiell gilt, dass mehr Variablen als irrelevant klassifiziert und entfernt werden, je größer α_{EA} gewählt wird. Es lässt sich aber eine intuitive Wahl des Schwellwerts durch folgende Überlegung angeben. Hierzu wird angenommen, dass die Systemstruktur des RFS die Nichtlinearität des Prozesses hinreichend gut abbilden kann. Berücksichtigen wir nun eine nicht relevante Größe, so führt dies zu einer Überanpassung des KRFS an die Trainingsdaten, da mehr Parameter zur Verfügung stehen als notwendig ist. Diese Überanpassung führt gleichzeitig zu einem Anstieg des Validierungsfehlers, wenn man voraussetzt, dass sich der Validierungsdatensatz hinreichend vom Trainingsdatensatz unterscheidet. Im Umkehrschluss ergibt sich dann als sinnvolle Wahl des Grenzwerts $\alpha_{EA} = 0$, d.h., eine Variable ist dann nicht relevant, wenn nach ihrem Ausschluss der Validierungsfehler sinkt. Dieser Wert hat sich auch in der Anwendung als sinnvoll herausgestellt.

Abschließend sei noch darauf hingewiesen, dass die Extraktion von Prozesswissen nicht auf die Zustands- und Eingangsvariablenauswahl beschränkt ist. Vielmehr kann der Ansatz auch auf die Extraktion anderer Formen von Prozesswissen erweitert werden, beispielsweise Informationen über das Systemverhalten in Bereichen des Zustandsraums oder über spezielles nichtlineares Systemverhalten, wie es in Abschnitt 5.2.1 beschrieben wurde. Darüber hinaus kann man obigen Algorithmus auch auf die Auswahl einzelner Koeffizienten des multilinearen Polynoms erweitern [169].

5.5 Simulationsbeispiele

Die Leistungsfähigkeit der in den vorangegangenen Abschnitten vorgestellten Algorithmen zur Modellbildung soll im Folgenden anhand zweier Simulationsbeispiele gezeigt werden. Anhand des Modells eines Bioreaktors wird gezeigt, wie aus Beispieldaten eine vollständige Regelbasis eines stark nichtlinearen dynamischen Systems erstellt werden kann. Die Auswahl der relevanten Eingangs- und Zustandsgrößen wird mittels eines höherdimensionalen Modells eines Rührkesselreaktors gezeigt.

5.5.1 Bioreaktor

Die Leistungsfähigkeit kontinuierlicher rekurrenter Fuzzy-Systeme zur Modellierung technischer Systeme soll am Beispiel eines Bioreaktors gezeigt werden [6]. Bioreaktoren stellen biochemische Prozesse dar, bei denen mit-

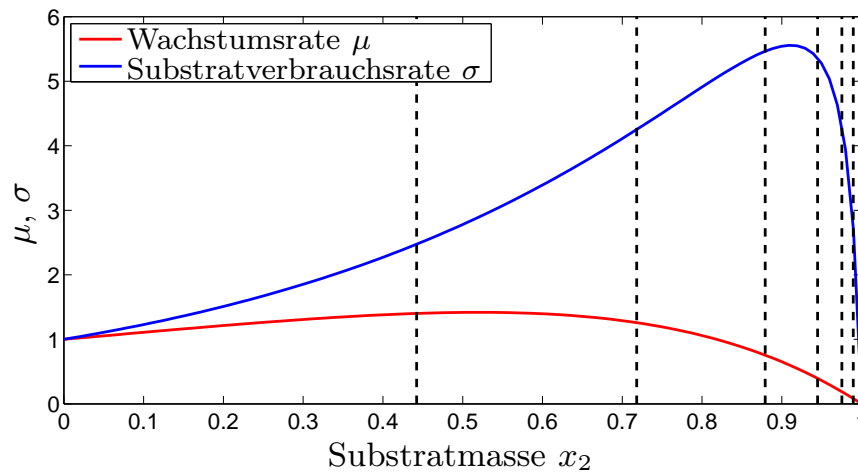


Abbildung 5.6: Typischer Verlauf von Wachstums- und Substratverbrauchsrate über der Substratmasse. Die senkrechten gestrichelten Linien markieren die optimierten Kernpositionen $s_{j_2}^{x_2}$.

hilfe von Bakterien oder Zellkulturen Zellbestandteile oder Stoffwechselprodukte gewonnen werden. Hierzu wird dem Reaktor kontinuierlich eine Nährstofflösung zugeführt, wodurch nach der biochemischen Reaktion kontinuierlich Zellen und Nährstoffe (Substrate) entnommen werden können. Die Modellierung derartiger Bioreaktoren erfolgt zumeist durch Aufstellen der Massenbilanzen [8]. Betrachtet man nur die normierten Zellmassen x_1 und Substratmassen x_2 , so lassen sich die folgenden Bilanzen

$$\begin{aligned}\dot{x}_1 &= \mu(x_2)x_1 - ux_1, \\ \dot{x}_2 &= \sigma(x_2)x_1 - ux_2\end{aligned}\tag{5.23}$$

herleiten. Hierbei ist u die Verdünnungsrate, die als Stellgröße dient, μ die Wachstumsrate und σ die Substratverbrauchsrate.

Im vorliegenden Benchmark-System (5.23) werden die Wachstums- und Verbrauchsrate (Abbildung 5.6) mit dem Substrat-Inhibition-Modell [8]

$$\begin{aligned}\mu(x_2) &= (1 - x_2)e^{x_2/\gamma}, \\ \sigma(x_2) &= \frac{1 + \beta}{1 + \beta - x_2}\mu(x_2)\end{aligned}$$

und den Modellparametern $\beta = 0,02$, $\gamma = 0,48$ nachgebildet. Im Allgemeinen ist die Modellierung solcher Bioreaktoren aber dadurch erschwert, dass die genaue Gestalt der Wachstums- und Verbrauchsrate nicht bekannt ist. Neben dem Substrat-Inhibition-Modell existieren eine Reihe weiterer Modelle [8, 15, 131, 138]. Die Verwendung von RFS zur Modellierung

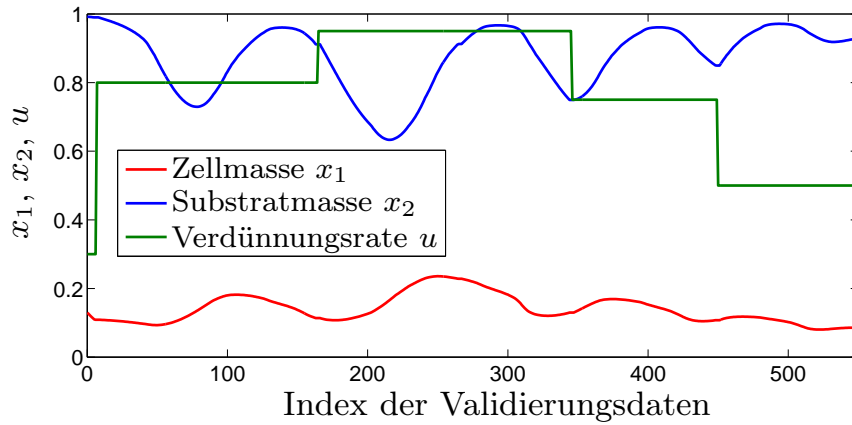


Abbildung 5.7: Verlauf der Validierungsdaten.

hat nun den Vorteil, dass Bioreaktoren modelliert werden können ohne Annahmen über Wachstumsmodelle treffen zu müssen. Zur Generierung von Trainingsdaten wird der Prozess mit einem amplitudenmodulierten Pseudo-Rausch-Binär-Signal (APRBS) im Bereich $u \in [0,3 \quad 1,1]$ angeregt. Bei der Wahl dieses Anregungsbereichs wurde darauf geachtet, dass insbesondere der Bereich hoher Substratkonzentrationen ausreichend angeregt wird, da vor allem hier stark nichtlineares Verhalten zu beobachten ist (vgl. Abbildung 5.6). Zusätzlich wurde ein Validierungsdatensatz generiert (Abbildung 5.7), mit dem die Generalisierungsfähigkeit des KRFS überprüft werden soll. Die Daten werden für die folgenden Beispiele als rauschfrei angenommen. Der zuvor beschriebene Algorithmus zur Strukturoptimierung wird nun auf den Trainingsdatensatz angewendet, zunächst ohne Beschränkung der Kernpositionsableitungen. Die Generalisierungsfähigkeit des optimierten Modells wird durch Beobachtung des Validierungsfehlers sichergestellt. Die Approximationsgüte wird mithilfe der Wurzel aus dem mittleren quadratischen Fehler (engl. root mean square error, RMSE) gemäß

$$\text{RMSE}_i = \sqrt{\frac{1}{N} \sum_{v=1}^N \left(\dot{x}_i^{(v)} - \hat{\dot{x}}_i^{(v)} \right)^2}$$

untersucht.

Der Strukturalgorithmus weist der Substratmasse x_2 sechs Kernpositionen zu, während die Zahl an Kernpositionen für die Zellmasse x_1 und die Verdünnungsrate u bei zwei bleibt. Dieses Ergebnis ist im Einklang mit der Systemdynamik des Bioreaktors (5.23). Die Nichtlinearität des Modells ist nämlich insbesondere durch die in Abbildung 5.6 dargestellten Wachstums-, und Substratverbrauchsrate bedingt, die nur von der Substratmasse ab-

hängen. Entsprechend werden mehr Kernpositionen für die Substratmasse benötigt, um die Nichtlinearitäten $\mu(x_2)$ und $\sigma(x_2)$ beschreiben zu können.

Zur Veranschaulichung der Funktionsweise der Strukturoptimierung wird Abbildung 5.6 um die Lage der sechs optimierten Kernpositionen für x_2 ergänzt. Da die Nichtlinearität hauptsächlich durch die Substratverbrauchsrate σ erzeugt wird, passt sich das durch die Kernpositionen entstehende Gitter (vgl. Abbildung 3.3) dieser an. Entsprechend liegen die Kernpositionen vermehrt im Bereich hoher Substratmassen.

Die Qualität der Modellierung wird in Abbildung 5.8 anhand der Verläufe der Ableitungen der Zellmasse und der Substratmasse des Trainingsdatensatzes gezeigt. Man erkennt, dass das KRFS eine sehr gute Übereinstimmung mit den Trainingsdaten aufweist (vgl. Tabelle 5.2). Die Generalisierungsfähigkeit des KRFS lässt sich anhand der Verläufe des Validierungsdatensatzes zur Einschnittsprädiktion in Abbildung 5.9 zeigen. Auch hier ist eine sehr gute Übereinstimmung zu beobachten.

Wie bereits angesprochen, ist die Interpretierbarkeit des gelernten Modells aufgrund der großen Zahl von 24 Ausgangszugehörigkeitsfunktionen kaum gegeben. Zur Verringerung der Zahl an Kernpositionsableitungen wird nun der in Abschnitt 5.3.2 beschriebene Algorithmus eingesetzt. Abbildung 5.10 zeigt, wie sich der RMSE über der Zahl an Kernpositionsableitungen verändert, wobei die Parameter jeweils mit QP optimiert wurden. Wie erwartet verschlechtert sich der RMSE bei abnehmender Zahl an Kernpositionsableitungen, sodass ein Kompromiss zwischen Interpretierbarkeit und Modellgüte gefunden werden muss. Bei entsprechender Einstellung der Parameter des ASK ergibt sich, dass auch mit jeweils 10 Kernpositionsableitungen in \dot{x}_1 und \dot{x}_2 eine zufriedenstellende Modellgüte erreicht werden kann. Die abschließende Partitionierung des Zustandsraums sowie die Kernpositionsableitungen sind in Abbildung 5.11 illustriert. Zum Vergleich wurde außerdem ein affin-lineares Modell identifiziert, das allerdings mit einem RMSE der Validierungsdaten von 0,0171 für \dot{x}_1 bzw. 0,0837 für \dot{x}_2 deutlich schlechter ist als das KRFS. Dies ist auch nachvollziehbar, da das KRFS ein Modell mit insgesamt fünf verschiedenen affin-multilinearen Funktionen darstellt und entsprechend mehr Parameter zur Verfügung stehen. Aufgrund des nichtlinearen Systemverhaltens reicht ein lineares Modell zur Beschreibung des Bioreaktors nicht aus.

Abbildung 5.12 zeigt die resultierende Regelbasis, mit der die dynamischen Eigenschaften des Bioreaktors linguistisch beschrieben werden können. Aus der Regelbasis lässt sich das Verhalten der bestimmenden Nichtlinearitäten, ausgedrückt durch Wachstums- und Substratverbrauchsrate, ablesen. Betrachtet man in der Regelbasis den Verlauf von \dot{x}_1 über der Sub-

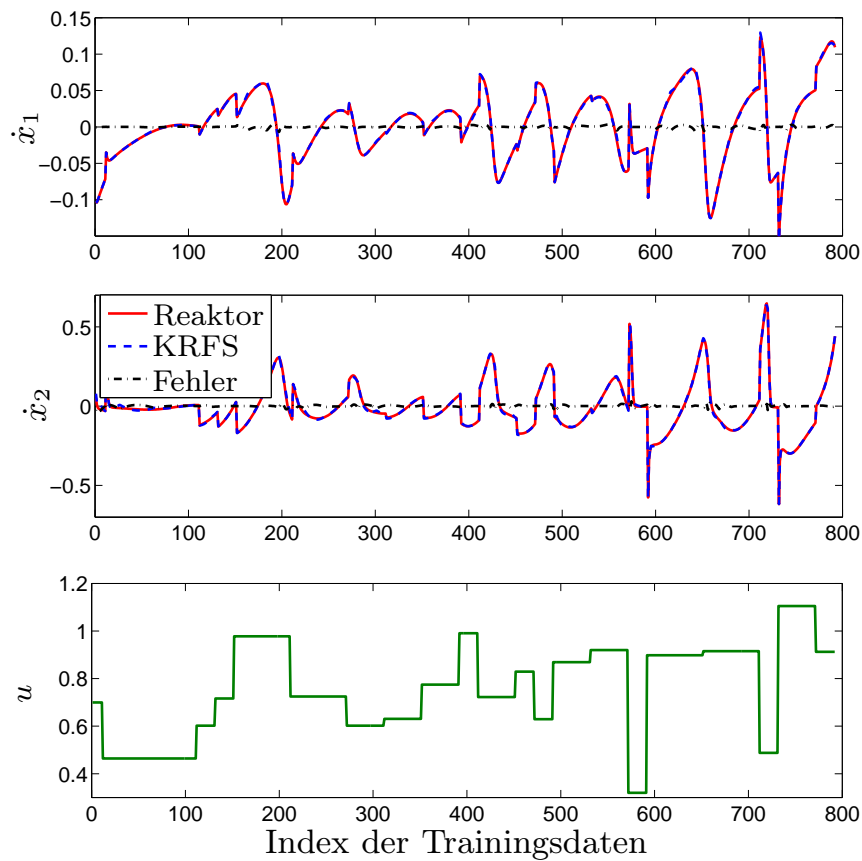


Abbildung 5.8: Vergleich der Zustandsableitungen \dot{x}_1 und \dot{x}_2 für den Trainingsdatensatz für eine seriell-parallele Modellauswertung.

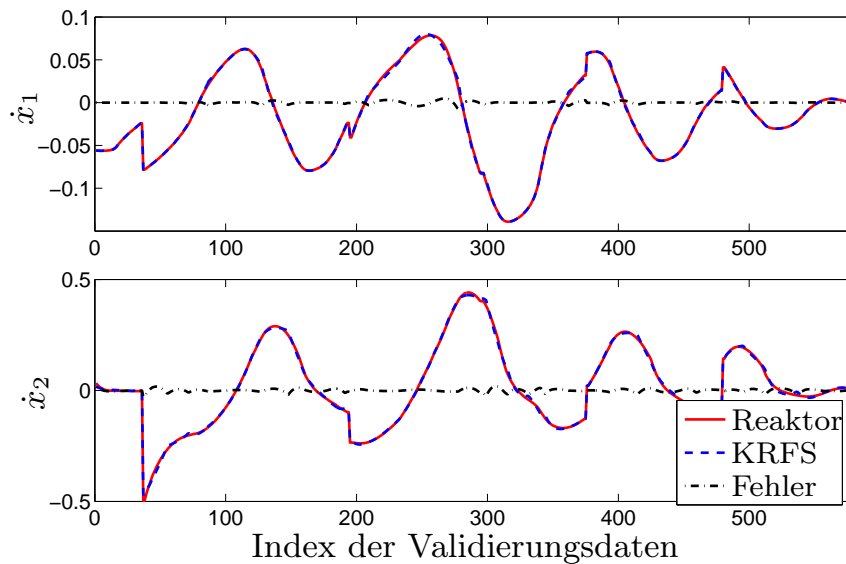


Abbildung 5.9: Vergleich der Zustandsableitungen \dot{x}_1 und \dot{x}_2 für den Validierungsdatensatz für eine seriell-parallele Modellauswertung.

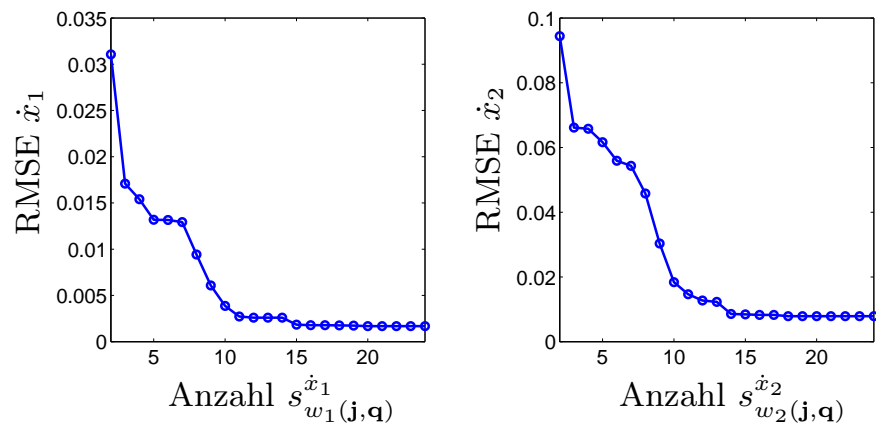


Abbildung 5.10: Entwicklung des RMSE über der Zahl an Kernpositionsableitungen für \dot{x}_1 (links) und \dot{x}_2 (rechts). Erwartungsgemäß nimmt der RMSE bei geringerer Zahl an Kernpositionsableitungen zu.

Tabelle 5.2: Ergebnisse der Modellierung des Bioreaktors. Angegeben ist jeweils der RMSE und der maximale Fehler.

	Trainingsfehler				Validierungsfehler			
	unbeschr.		beschr.		unbeschr.		beschr.	
	RMSE	MAX	RMSE	MAX	RMSE	MAX	RMSE	MAX
\dot{x}_1	0,0014	0,0049	0,0043	0,0053	0,0017	0,0151	0,0038	0,0135
\dot{x}_2	0,0066	0,0224	0,0184	0,0293	0,0092	0,0871	0,0183	0,0385

stratmasse x_2 , so erkennt man in Übereinstimmung mit dem Verlauf der Wachstumsrate (Abbildung 5.6) eine kontinuierliche Abnahme, die mit x_1 skaliert wird. Das inhibierende Verhalten der Substratverbrauchsrate wird insbesondere aus der Regelbasis von \dot{x}_2 deutlich. Betrachtet man hierzu die Regeln für $x_1 = \text{“g“}$, so erkennt man einen Anstieg der linguistischen Werte in \dot{x}_2 bis $x_2 = \text{“mg“}$ und einen deutlichen Abfall für größere Substratmassen. Dies entspricht exakt dem aufgrund der Lage der Kernpositionen in Abbildung 5.6 zu erwartenden Verhalten. Ist dagegen $x_1 = \text{“k“}$, tritt kein Sprung in den linguistischen Werten auf. Da x_1 multiplikativ wirkt, hat die Nichtlinearität angesichts der hohen Verdünnungsrate nur geringen Einfluss und tritt in der Regelbasis nicht hervor.

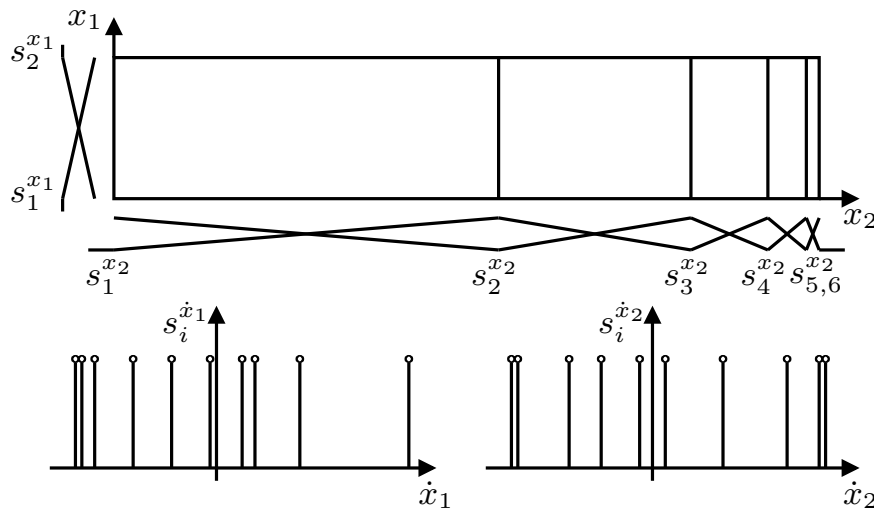


Abbildung 5.11: Partitionierung des Zustandsraums und Kernpositionsableitungen der Zustandsableitungen.

		Ableitung der Zellmasse \dot{x}_1					
		x_2					
x_1	$u=k$	sk	k	mk	mg	g	sg
	k	7	7	6	6	5	5
	g	10	9	8	6	4	4

		Ableitung der Substratmasse \dot{x}_2					
		x_2					
x_1	$u=k$	sk	k	mk	mg	g	sg
	k	5	5	5	5	5	4
	g	6	9	10	10	8	6

Abbildung 5.12: Regelbasis, mit der die Dynamik des Bioreaktors linguistisch beschrieben werden kann. Hierbei ist sk=„sehr klein“, k=„klein“, mk=„mittel klein“, mg=„mittel groß“, g=„groß“, sg=„sehr groß“. Die Zahlen von 1-10 kennzeichnen aufsteigend linguistische Werte für die Ableitung der Zustände.

5.5.2 Rührkesselreaktor

Nachdem im vorangegangenen Abschnitt die Leistungsfähigkeit der Struktur- und Parameteroptimierung vorgestellt wurde, soll nun die Leistungsfähigkeit der Variablenselektion gezeigt werden [172]. Als Beispiel wird die Modellierung eines Rührkesselreaktors [105, 160] gezeigt. Der Aufbau des Rührkesselreaktors ist in Abbildung 5.13 zu sehen. Dem Rührkessel wird kontinuierlich mit einem Eingangsstoff A mit Konzentration c_{in} und Temperatur T_{in} gespeist. Dieser Stoff A reagiert innerhalb des Reaktors in den Stoff B, wobei als Beiprodukte die Stoffe C und D entstehen. Diese

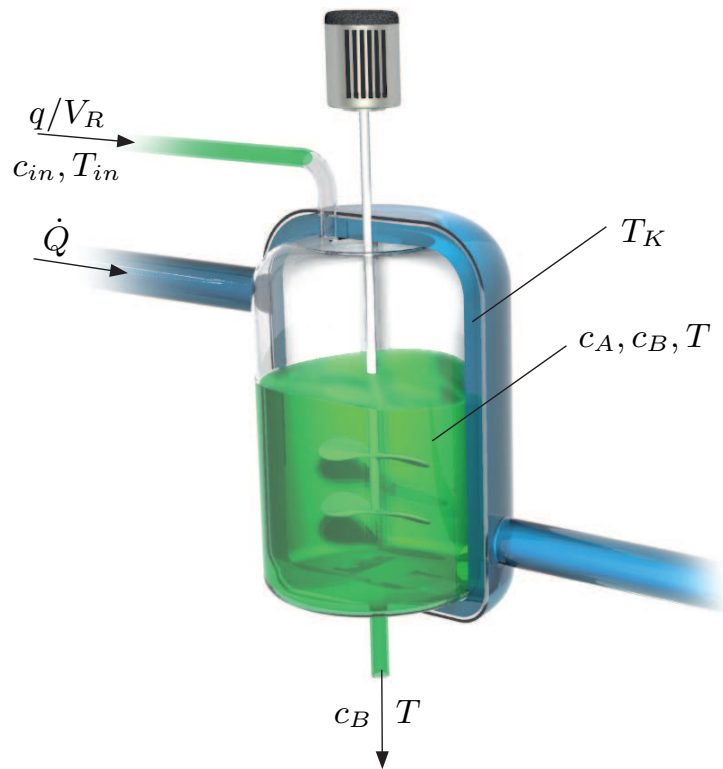


Abbildung 5.13: Aufbau des Rührkesselreaktors mit Mantelkühlung.

Stoffe werden am Boden entnommen. Die Temperatur im Reaktor kann mithilfe einer Mantelkühlung reguliert werden.

Durch Bildung der Massenbilanzen für die Konzentrationen c_A und c_B erhält man

$$\dot{c}_A = -k_1(T)c_A - k_3(T)c_A^2 + (c_{in} - c_A)u_1, \quad (5.26)$$

$$\dot{c}_B = k_1(T)(c_A - c_B) - c_B u_1, \quad (5.27)$$

wobei $k_1(T)$ und $k_3(T)$ temperaturabhängige Konstanten sind, $c_{in} = 5,1 \text{ mol/l}$ ist die Konzentration des Edukts A im Zulauf und $u_1 = \frac{q}{V_R}$ ist die auf das Reaktorvolumen V_R bezogene Zuflussrate, die die erste Eingangsvariable darstellt. Des Weiteren führen die Enthalpiebilanzen für die Temperaturen des Reaktors T und des Kühlmantels T_k auf

$$\begin{aligned} \dot{T} &= -k_1(T)(\alpha_1 c_A + \alpha_2 c_B) - \alpha_3 k_3(T)c_A^2 \\ &\quad + \alpha_4(T_k - T) + (T_{in} - T)u_1, \\ \dot{T}_k &= \alpha_5(T - T_k) + \alpha_6 u_2, \end{aligned} \quad (5.28)$$

mit der Zulauftemperatur $T_{in} = 104,9^\circ\text{C}$, den Konstanten $\alpha_1 = 1,48^\circ\text{Cm}^3/\text{mol}$, $\alpha_2 = 3,91^\circ\text{Cm}^3/\text{mol}$, $\alpha_3 = 14,74^\circ\text{Cm}^3/\text{mol}$, $\alpha_4 = 30,83/\text{h}$,

$\alpha_5 = 86,69/\text{h}$ und $\alpha_6 = 0,1^\circ\text{C}/\text{kJ}$ sowie der Kühlleistung $u_2 = \dot{Q}$, die die zweite Eingangsvariable darstellt. Die temperaturabhängigen Konstanten $k_1(T)$ und $k_2(T)$ werden durch Arrhenius-Funktionen

$$k_i(T) = k_{i0} \cdot 10^9 e^{\frac{-E_i}{T/^\circ\text{C} + 273,15}}, \quad i = 1, 2,$$

mit $k_{10} = 1287 \text{ h}^{-1}$, $k_{20} = 9,03 \text{ m}^3\text{mol} \cdot \text{h}^{-1}$, $E_1 = 9758,3$ und $E_2 = 8560$ modelliert.

Definiert man abschließend den Zustandsvektor $\mathbf{x} = [c_A \ c_B \ T \ T_k]^T$, so erhält man das System

$$\dot{\mathbf{x}} = \begin{pmatrix} f_1(x_1, x_3, u_1) \\ f_2(x_1, x_2, x_3, u_1) \\ f_3(x_1, x_2, x_3, x_4, u_1) \\ f_4(x_3, x_4, u_2) \end{pmatrix}, \quad (5.29)$$

wobei $f_i(\mathbf{x}, \mathbf{u})$ die Zustandsgleichungen (5.26) - (5.28) sind.

Als erster Schritt zur Nachbildung des chemischen Reaktors durch ein KRFS wird zunächst untersucht, inwieweit qualitatives Wissen über den Prozess vorliegt. Dies ist bei chemischen Prozessen im Regelfall durch die im Reaktor ablaufenden Reaktionen gegeben. Man weiß dann, welcher Stoff mit welchem Stoff reagiert und entsprechende Produkte und Beiprodukte liefert. Diese Information kann durch Massenbilanzen wie in Gl. (5.26)-(5.27) beschrieben werden. Genaues Wissen hat man hierbei über die beteiligten Prozessgrößen, dagegen sind die Reaktionskinetiken oftmals unbekannt. Hieraus ergibt sich schließlich die qualitative Information, dass das System eingangs-affin ist. Außerdem geht man davon aus, dass der Zusammenhang zwischen Reaktor- und Manteltemperatur durch einen linearen Wärmeübergang hinreichend genau beschrieben werden kann. Diese Informationen werden mithilfe der in Abschnitt 5.2.1 beschriebenen Nebenbedingungen berücksichtigt. Wie Gl. (5.29) zu entnehmen ist, gehen nicht alle Zustände und Eingänge in die Zustandsgleichungen ein, sodass die Anwendung des Algorithmus zur Variablenauswahl aus Abschnitt 5.4 sinnvoll ist, um weitere qualitative Informationen zu gewinnen.

Hierzu werden wiederum Datensätze durch Anregung mittels eines APRBS-Signals erzeugt, wodurch sich 5000 Trainings- und 1000 Validierungsdatenpaare ergeben (Abbildungen B.1-B.3 im Anhang). Zunächst wird der Strukturalgorithmus aus Abschnitt 5.3.1 angewendet. Es ergibt sich, dass zwei Zugehörigkeitsfunktionen für die Zustände x_1 , x_2 , x_4 und Eingänge u_1 , u_2 sowie fünf Zugehörigkeitsfunktionen für x_3 hinreichend

Tabelle 5.3: Ergebnisse der lokalen Variablenauswahl. Für die Ableitung \dot{x}_3 werden der Zustand x_2 und Eingang u_1 nur in 3 von 4 Hyperquadranten als relevant eingestuft, während der Eingang u_2 nur in einem Hyperquadranten als relevant eingestuft wird.

	x_1	x_2	x_3	x_4	u_1	u_2
\dot{x}_1	4/4	0/4	4/4	0/4	4/4	0/4
\dot{x}_2	4/4	4/4	4/4	0/4	4/4	0/4
\dot{x}_3	4/4	3/4	4/4	4/4	3/4	1/4
\dot{x}_4	0/4	0/4	4/4	4/4	0/4	4/4

Tabelle 5.4: Vergleich des RMSE für seriell-parallele Modellauswertung sowie beschränkte und unbeschränkte Optimierung. Zusätzlich ist die Zahl an freien Optimierungsparametern angegeben.

	Trainingsfehler		Validierungsfehler		Freie Parameter	
	unbeschr.	beschr.	unbeschr.	beschr.	unbeschr.	beschr.
\dot{x}_1	0,039	0,201	0,452	0,219	160	20
\dot{x}_2	0,007	0,059	0,139	0,057	160	40
\dot{x}_3	0,337	1,534	3,712	1,675	160	80
\dot{x}_4	≈ 0	≈ 0	≈ 0	≈ 0	160	20

für eine genaue Modellierung sind. Somit ergeben sich 160 Regeln für jede Zustandsableitung. Anschließend wird die Variablenauswahl durchgeführt. Die Ergebnisse sind in Tabelle 5.3 zusammengefasst. Offenbar wählt der Algorithmus genau die Variablen aus, die die Ableitung der Zustandsvariablen direkt beeinflussen, was die Anwendbarkeit des Ansatzes unterstreicht.

Abschließend werden die Kernpositionsableitungen unter Berücksichtigung der zuvor ermittelten relevanten Eingänge und Zustände optimiert. Ein Vergleich der Trainings- und Validierungsfehler bei Berücksichtigung aller Größen und ausschließlicher Berücksichtigung der relevanten Größen zeigt Tabelle 5.4. Außerdem ist die Zahl an freien Parametern angegeben. Wie erwartet sind die Trainingsfehler bei Berücksichtigung aller Zustände und Eingänge deutlich niedriger als mit Beschränkungen, da deutlich mehr

Tabelle 5.5: Vergleich des maximalen Fehlers für seriell-parallele Modellauswertung sowie beschränkte und unbeschränkte Optimierung. Zusätzlich ist die Zahl an freien Optimierungsparametern angegeben.

	Trainingsfehler		Validierungsfehler		Freie Parameter	
	unbeschr.	beschr.	unbeschr.	beschr.	unbeschr.	beschr.
\dot{x}_1	0,551	1,207	9,108	0,841	160	20
\dot{x}_2	0,105	0,705	5,839	0,28	160	40
\dot{x}_3	2,47	12,765	166,98	4,881	160	80
\dot{x}_4	≈ 0	≈ 0	≈ 0	≈ 0	160	20

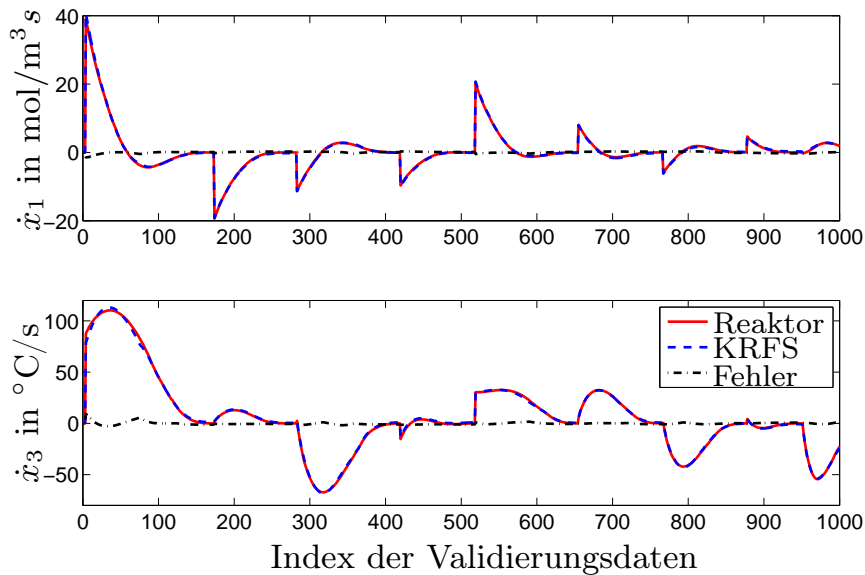


Abbildung 5.14: Vergleich der Verläufe der Zustandsableitungen \dot{x}_1 und \dot{x}_3 für den Validierungsdatensatz für eine seriell-parallele Modellauswertung. Das KRFS mit relevanten Eingängen und Zuständen kann die Dynamik des Systems auch für unbekannte Daten sehr gut nachbilden.

Optimierungsparameter zur Verfügung stehen. Dagegen sind die Validierungsfehler bei Beschränkungen deutlich niedriger. Auffällig ist auch, dass die RMSE der Trainings- und Validierungsdatensätze bei unbeschränkter Optimierung um Größenordnungen auseinander liegen. Dies deutet auf eine inakzeptable Überanpassung an den Trainingsdatensatz hin. Dagegen liegen die RMSE bei beschränkter Optimierung nahe beieinander, sodass sich ein Modell mit guter Generalisierungsfähigkeit ergibt. Ähnliche Ergebnisse

zeigen sich auch bei der Betrachtung des maximalen Fehlers (Tabelle 5.5). Durch ausschließliche Berücksichtigung der relevanten Größen erhöht sich also die Generalisierungsfähigkeit deutlich. Abbildung 5.14 zeigt die Verläufe des Validierungsdatensatzes im Vergleich zum RFS für die erste und dritte Zustandsableitung, die eine sehr hohe Übereinstimmung aufweisen. Für die anderen Zustandsableitungen sind die Verläufe ähnlich. Zum Vergleich wurde auch hier ein affin-lineares Modell identifiziert. Der RMSE der Validierungsdaten von 4,1284 für \dot{x}_1 , 0,8324 für \dot{x}_2 bzw. 14,7678 für \dot{x}_3 ist deutlich schlechter als der des KRFS. Die vierte Zustandsableitungen kann dagegen erwartungsgemäß genau modelliert werden, da es sich hierbei um einen linearen Zusammenhang handelt. Wie bereits beim Bioreaktor kann das nichtlineare Systemverhalten durch ein lineares Modell nicht ausreichend nachgebildet werden.

5.6 Zusammenfassung und Fazit

Dieses Kapitel beschäftigte sich mit der Modellierung dynamischer Systeme mithilfe zeitkontinuierlicher RFS. Hierzu wurden Algorithmen vorgestellt, die sowohl die Parameter des KRFS als auch die Struktur, also die Zahl an Zugehörigkeitsfunktionen und die relevanten Eingangs- und Zustandsgrößen, bestimmen. Des Weiteren wurde gezeigt, wie mithilfe geeigneter Nebenbedingungen verschiedene Formen von Vorwissen in den Modellierungsprozess integriert werden können, sodass neben Messdatensätzen möglichst viele Informationen eingebracht werden können. Problematisch bei der Modellierung mit RFS ist der exponentielle Anstieg der Zahl an Regeln bei höherdimensionalen Systemen, da zum einen die Interpretierbarkeit des KRFS verringert wird und zum anderen durch die hohe Zahl an Optimierungsparametern die Generalisierungsfähigkeit des Modells abnimmt. Dies lässt sich durch die Definition verschiedener Nebenbedingungen mindern. Zum einen können durch Vorwissen oder datengestützt Nebenbedingungen definiert werden, die die Zahl an Optimierungsparametern sinnvoll verringern. Zum anderen können redundante Regeln durch Nebenbedingungen entfernt werden. Im Ergebnis erhält man ein vollständiges Zustandsraummodell des Systems, dessen Dynamik durch eine Regelbasis gegeben und somit linguistisch interpretierbar ist. Dies ermöglicht im Vergleich mit reinen Black-Box Modellen die Plausibilisierung des erhaltenen Modells nach der Modellbildung und erweitert so die Kenntnisse über das System.

Ein Nachteil der vorgestellten Modellierungsverfahren ist die Annahme, dass alle Zustände gemessen werden können, was in der Praxis nicht immer

vorausgesetzt werden kann. Allerdings ist zu beachten, dass die Zustände nur zur Aufstellung des Modells benötigt werden. Wird das Modell später online verwendet, beispielsweise zur Fehlerdiagnose oder Regelung, ist die Kenntnis der Ausgangsgrößen ausreichend, wenn auch die Ausgangsfunktion modelliert wird. Außerdem existieren eine Reihe von technischen Prozessen, bei denen entweder die Zustände gemessen werden (siehe z.B. Brennstoffzelle [49]) oder während der Entwicklung Prototypen existieren, die komplett vermessen werden. Diese Daten können dann zur Modellierung verwendet werden.

Der obige Nachteil lässt sich beheben, wenn nur Ein-/Ausgangsdaten zur Modellierung verwendet werden. Die Optimierung der Parameter hängt dann allerdings zusätzlich von der Zeit ab und führt auf ein Gradientenverfahren mit Backpropagation-Through-Time [207]. Außerdem erhöht sich die Zahl an nichtlinearen Parametern um die Zahl an Kernpositionsableitungen, was zu einem erheblich vergrößerten Suchraum mit vielen lokalen Minima führt. Hieraus folgt, dass das Backpropagation-Through-Time nur dann erfolgsversprechend anwendbar ist, wenn sehr gute Initialwerte für die Parameter bekannt sind. Es muss also auch quantitativ sehr genaues Vorwissen vorhanden sein. Ist solch genaues Wissen nicht vorhanden, so kann der Vorteil RFS, nämlich die Möglichkeit, interne Zusammenhänge über Zustandsgrößen nachbilden zu können, nicht mehr ausgenutzt werden. Dann eignen sich reine Black-Box-Modelle, wie Neuronale Netze oder Neuro-Fuzzy-Systeme, deutlich besser zur Modellbildung.

6 Modellierung mit zeitdiskreten rekurrenten Fuzzy-Systemen

Bisherige Anwendungsgebiete für DRFS mit Rückkopplungskorrespondenz sind die Mustererkennung, zu der sie auch industriell eingesetzt werden [2, 99], sowie eine Verkehrssimulation [50]. In diesem Abschnitt soll nun gezeigt werden, wie Modelle dynamischer Systeme mit DRFS, die in ihrer Funktionsweise Automaten ähneln, erstellt werden können. Die Optimierung der Parameter eines DRFS mithilfe evolutionärer Algorithmen wurde bereits in [59] vorgestellt. Allerdings wurden die DRFS in dieser Arbeit so modifiziert, dass die Rückkopplungskorrespondenz nicht mehr erfüllt sein muss, was die Optimierung deutlich vereinfacht. Dagegen beschäftigt sich dieses Kapitel nicht nur mit der Optimierung der Parameter von DRFS mit Rückkopplungskorrespondenz, sondern konzentriert sich vor allem auf die Optimierung der Struktur in Form der Regelbasis. Das Ziel ist hierbei insbesondere, Modelle zu entwickeln, die den Prozess qualitativ genau nachbilden.

6.1 Einführung und Problemdefinition

Im Gegensatz zur im vorigen Kapitel vorgestellten Modellierung dynamischer Systeme mittels KRFS, die die Entwicklung von interpretierbaren Approximationsmodellen zum Ziel hatte, widmet sich dieses Kapitel vornehmlich der *qualitativen Modellierung* dynamischer Systeme. Dies wird zum einen durch die Automatenrepräsentation DRFS motiviert, die bereits ein qualitatives Modell darstellt, zum anderen durch die im Vergleich zu KRFS deutlich verringerte Zahl an freien Parametern, die eine genaue *quantitative Modellierung* erschweren.

Zur qualitativen Modellierung sind eine Reihe von Verfahren entwickelt worden, die sich grob in qualitatives Schließen und naive Physik sowie hybride bzw. ereignisdiskrete Systeme aufteilen lassen [125]. Verfahren der

ersten Gruppe teilen das zu modellierende System entweder in Komponenten auf oder beschreiben das System durch physikalische Beschränkungen. Das Verhalten und die Interaktion der Module oder Beschränkungen wird entweder durch Regeln mit binären Zugehörigkeitsfunktionen (qualitatives Schließen [206]) oder durch einfache physikalische Zusammenhänge (naive Physik [52]) beschrieben. Verfahren der zweiten Gruppe basieren auf einer Quantisierung des Eingangsraums, d.h., für die Systemgrößen werden qualitative Intervalle festgelegt [125]. Das resultierende System wird dann als ereignisdiskretes System interpretiert. Je nach Auslegung können die Systeme deterministisches oder nichtdeterministisches Verhalten aufweisen und durch Automaten, Petri-Netze oder gerichtete Graphen beschrieben werden [127].

Aufgrund des qualitativen Verhaltens werden solche Modelle weniger zur genauen Prozessprädiktion als vielmehr als abstrahierte Systembeschreibungen in der Überwachungsebene oder Prozessdiagnose eingesetzt. Ein Vorteil diskreter RFS mit Rückkopplungskorrespondenz im Vergleich zu obigen Verfahren besteht darin, dass neben der reinen qualitativen Nachbildung durch die Interpolation zwischen den Regeln in gewissem Umfang auch eine quantitative Nachbildung erfolgt. Entsprechend können DRFS auch in eingeschränkter Form zur Prädiktion verwendet werden.

Der beschriebene Fokus auf die qualitative Modellierung hat auch Auswirkungen auf die Gestaltung der Datensätze. Zur Modellierung mit KRFS bzw. DRFS ohne Rückkopplungskorrespondenz wurden mithilfe von geeigneten Anregungssignalen kontinuierliche Datensätze mit niedrigen Abtastzeiten verwendet. Dies ist allerdings zur qualitativen Modellierung nicht sinnvoll, da nicht die Zusammenhänge zwischen kurz aufeinander folgenden Messdaten abgebildet werden sollen. Vielmehr sollen qualitative Werte der Zustands- und Eingangsgrößen wie „klein“, „mittel“ und „groß“ miteinander in Zusammenhang gesetzt werden. Folglich benötigt man Datensätze, die Muster von obigen qualitativen Werten enthalten. Um dies zu erreichen, werden zur Generierung der Trainings- und Validierungsdaten deutlich größere Abtastzeiten verwendet, als bei der in Kapitel 5 vorgestellten Modellierung mit KRFS. Die Wahl dieser Abtastzeit ist allerdings nicht trivial, da sich das ergebene DRFS bei verschiedenen Abtastzeiten deutlich unterscheiden kann. Hierzu betrachte man als einfaches Beispiel die Regel:

Wenn $x(k)$ = „klein“ und $u(k)$ = „groß“, dann $x(k+1)$ = „mittel“.

Verdoppelt man nun die Abtastzeit so ergibt sich eventuell als linguistischer Wert im nächsten Zustand nicht mehr $x(k+1)$ = „mittel“, sondern

$x(k+1)$ = „groß“. Entsprechend ist die Ausprägung des erhaltenen DRFS und damit des qualitativen Modells also auch von der Abtastzeit abhängig. Gleichzeitig sollte die Abtastzeit so gewählt werden, dass die Zeitkonstanten, die durch das Modell wiedergegeben werden sollen, auch erfasst werden. Bei einem qualitativen Modell kann dies allerdings bedeuten, dass schnelle Zeitkonstanten nicht erfasst werden sollen und nur das langsame Systemverhalten abgebildet werden soll.

Nach dieser Einführung soll nun das Problem zur Modellierung dynamischer Prozesse durch DRFS mit Rückkopplungskorrespondenz formuliert werden. Insgesamt werden zwei Teilprobleme betrachtet. Das erste Problem besteht darin, bei bestehender Partitionierung des Eingangsraums die Regelbasis, also die Zuordnung von linguistischen Werten der Ausgangsgrößen zu den Regelprämissen der Eingangsgrößen, zu identifizieren. Das zweite Problem ist dann die Optimierung der kontinuierlichen Parameter, d.h. der Kernpositionen der Eingänge und Ausgänge. Bei beiden Problemen ist zu beachten, dass die Menge der linguistischen Ausgangswerte aufgrund der Rückkopplungskorrespondenz mit der Menge der linguistischen Eingangswerte identisch ist. Gleiches gilt für die Lage der zugehörigen Kernpositionen. Zusammenfassend lässt sich das folgende allgemeine Modellierungsproblem formulieren:

Modellierungsproblem 6.1. Seien N Datenpaare $(\mathbf{x}(k), \mathbf{u}(k), \mathbf{x}(k+1))$ gegeben. Dann

1. bestimme die Regelbasis des DRFS, sodass jeder Regel eine Ausgangszugehörigkeitsfunktion aus der Menge der möglichen Ausgangszugehörigkeitsfunktionen zugeordnet ist und
2. optimiere die Werte der freien Parameter des DRFS,

sodass der mittlere quadratische Fehler zwischen gemessenem Zustand $x_i^{(v)}(k+1)$ und durch das DRFS gegebenem Zustand $\hat{x}_i^{(v)}(k+1)$

$$\frac{1}{2} \sum_{v=1}^N (x_i^{(v)}(k+1) - \hat{x}_i^{(v)}(k+1))^2$$

minimiert wird.

Bei der Identifikation der Regelbasis handelt es sich um ein kombinatorisches Optimierungsproblem, da jeder Regel eine Zugehörigkeitsfunktion aus einer begrenzten Menge möglicher Zugehörigkeitsfunktionen zugeordnet werden soll. Solch ein kombinatorisches Optimierungsproblem

kann üblicherweise als ganzzahliges Optimierungsproblem formuliert werden [51, 191, 209]. Das zweite Teilproblem ist dagegen, ähnlich wie bei der Optimierung der Parameter von KRFS, ein nichtlineares Optimierungsproblem. Die Ausgangskernpositionen sind allerdings keine linearen Parameter mehr, da sie aufgrund der Rückkopplungskorrespondenz mit den Kernpositionen der Zustände übereinstimmen. Fasst man beide Teilprobleme zusammen, so ergibt sich ein nichtlineares gemischt-ganzzahliges Optimierungsproblem (NGGOP).

Es gibt nun zwei denkbare Möglichkeiten, dieses NGGOP zu lösen. Die erste besteht darin, das NGGOP direkt mit Methoden der gemischt-ganzzahligen Optimierung zu lösen. Die bisher entwickelten Verfahren zur Lösung von NGGOP fordern allerdings zumeist, dass das kontinuierliche Optimierungsproblem, das nach der Relaxierung der ganzzahligen Parameter zu kontinuierlichen Parametern entsteht, konvex ist [51]. Diese Forderung ist aber bei obigem Problem nicht erfüllt. Hintergrund der Forderung nach Konvexität ist, dass die Verfahren das Problem mithilfe von Branch-and-Bound Techniken lösen [51]. Das bedeutet, dass das Problem durch Festsetzen oder Relaxieren einiger ganzzahliger Parameter nach und nach in kontinuierliche Optimierungsprobleme überführt wird, deren Lösungen obere und untere Grenzen des Originalproblems darstellen. Durch die schrittweise Relaxierung entsteht eine baumartige Struktur. Erhält man nun im ersten Teilbaum eine untere Grenze, die oberhalb der oberen Grenze des zweiten Teilbaums liegt, so kann man ausschließen, dass der erste Teilbaum das globale Minimum enthält. Er wird deshalb nicht weiter betrachtet. Dies gilt aber nur dann, wenn die relaxierten Probleme konvex sind, denn nur dann konvergieren sie auch in ein globales Minimum. Ist das relaxierte Problem dagegen nicht konvex, so kann dessen Lösung auch ein (schlechtes) lokales Minimum sein, was zu nicht verlässlichen oberen und unteren Grenzen führt. In diesem Fall kann der Teilbaum mit dem globalen Optimum abgeschnitten werden, weil ein Teilproblem in ein Nebenminimum konvergiert ist. Eine erfolgreiche Lösung des obigen NGGOP mit dieser Methodik ist somit kaum möglich.

Die zweite Möglichkeit, die auch im Rahmen dieser Arbeit genutzt wird, besteht darin, die beiden Teilprobleme getrennt voneinander zu lösen. Dies führt auf ein ganzzahliges quadratisches sowie ein nichtlineares kontinuierliches Optimierungsproblem, für das jeweils Lösungsmethoden existieren. Erkauft wird dies durch eine gewisse Konservativität der Lösung, da das globale Optimum unter Umständen nicht gefunden wird. Dies liegt daran, dass bei der Lösung eines der beiden Teilprobleme die Parameter des

jeweils anderen Optimierungsproblem festgesetzt werden. Es werden also nicht alle Freiheitsgrade gleichzeitig optimiert.

Die schrittweise Lösung hat den Vorteil, dass sich die Schritte sehr gut in den Kontext der qualitativen und quantitativen Modellbildung, wie er in [49] beschrieben ist, einordnen lassen. Im ersten Schritt, der Bestimmung der Regelbasis mittels ganzzahliger Optimierung, wird das qualitative Modell bestimmt. Es wird also bestimmt, welche qualitativen Zustandsgrößen aus einer Kombination von qualitativen Eingangs- und Zustandsgrößen folgen. Mit den Ergebnissen lässt sich bereits ein linguistischer Zustandsgraph, wie in Abbildung 3.5 gezeigt, ableiten. Zur qualitativen Modellbildung reicht eine ungefähre Kenntnis der Kernpositionen, die durch Vorwissen festgelegt ist, aus. Die nachfolgende Optimierung der Kernpositionen quantifiziert dann das qualitative Modell, indem die Parameter so optimiert werden, dass sie den Prozess im Sinne eines quadratischen Gütekriteriums optimal nachbilden.

Darüber hinaus gibt es für DRFS neben der Modellbildung weitere Anwendungen, bei denen die Kernpositionen von vornherein feststehen, sodass eine Parameteroptimierung nicht nötig ist. Beispiele sind die sequentielle Mustererkennung und die in Abschnitt 7.3.2 noch genauer beschriebene dynamische Fehlerisolation, bei denen die Kernpositionen vor der Optimierung bekannt sind. Dann liegt das ausschließliche Augenmerk auf der Identifikation der Regelbasis.

Im Folgenden wird nun zunächst das ganzzahlige Optimierungsproblem zur Identifikation der Regelbasis hergeleitet. Anschließend wird die Optimierung der Parameter betrachtet.

6.2 Identifikation der Regelbasis mit ganzzahliger Optimierung

Dieser Abschnitt stellt die Lösung von Teilproblem 1 des Modellierungsproblems 6.1, also die Identifikation der Regelbasis, vor. Um das abstrakt formulierte Optimierungsproblem in ein analytisches Optimierungsproblem zu überführen, wird die Systemdarstellung diskreter rekurrenter Fuzzy-Systeme zunächst analysiert und modifiziert. Gemäß Abschnitt 3.2 ist die

Systemdarstellung RFS in der Komponente x_i des Zustands durch

$$\begin{aligned} x_i(k+1) &= \sum_{\mathbf{j}, \mathbf{q}} s_{w_i(\mathbf{j}, \mathbf{q})}^{x_i} \prod_{i=1}^n \mu_{j_i}^{x_i}(x_i) \prod_{p=1}^m \mu_{q_p}^{u_p}(u_p) \\ &= \sum_{\mathbf{j}, \mathbf{q}} s_{w_i(\mathbf{j}, \mathbf{q})}^{x_i} \Xi_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}, \mathbf{u}). \end{aligned} \quad (6.1)$$

gegeben, wobei in der letzten Umformung der Ausdruck $\Xi_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}, \mathbf{u})$ die Prämisse einer Regel darstellt. Da angenommen wird, dass die Eingangspartitionierung bekannt ist, ist auch $\Xi_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}, \mathbf{u})$ für jedes Messdatenpaar bekannt. Der Ausdruck $s_{w_i(\mathbf{j}, \mathbf{q})}^{x_i}$ bezeichnet die der entsprechenden Prämisse zugeordnete Ausgangszugehörigkeitsfunktion. Gemäß der Rückkopplungskorrespondenz ist die Menge der zur Verfügung stehenden Ausgangszugehörigkeitsfunktionen des Zustands x_i gleich der Menge der Kernpositionen für x_i . Über die Zuordnung von Ausgangszugehörigkeitsfunktionen zu Regelprämissen wird dann die Regelbasis bestimmt. Diese Zuordnung stellt mathematisch ein kombinatorisches Problem dar, da jeder Regel eine Zugehörigkeitsfunktion aus einer begrenzten Menge möglicher Zugehörigkeitsfunktionen zugeordnet werden soll. Wie bereits erwähnt, lässt sich hieraus zumeist ein ganzzahliges Optimierungsproblem formulieren [51, 209].

Um dieses herzuleiten, wird zunächst der Vektor der möglichen Ausgangszugehörigkeitsfunktionen definiert

$$\mathbf{s}^{x_i} = \begin{bmatrix} s_1^{x_i} & \dots & s_{r_x(i)}^{x_i} \end{bmatrix}^T \in \mathbb{R}^{r_x(i) \times 1}, \quad (6.2)$$

wobei $r_x(i)$ die Anzahl der Kernpositionen von x_i bezeichnet. Der Vektor (6.2) enthält somit alle Kernpositionen von x_i . Außerdem werden Vektoren

$$\mathbf{a}_{w_i} = \begin{bmatrix} a_1 & \dots & a_{r_x(i)} \end{bmatrix}^T \in \{0, 1\}^{r_x(i) \times 1},$$

definiert, deren Einträge a_j binäre Variablen sind. Bei entsprechender Wahl der binären Variablen gilt dann

$$s_{w_i}^{x_i} = \mathbf{a}_{w_i} \mathbf{s}^{x_i}. \quad (6.3)$$

Offenbar stellen die Vektoren \mathbf{a}_{w_i} Selektionsvektoren dar, die für eine Regel eine Ausgangszugehörigkeitsfunktion aus dem Vektor \mathbf{s}^{x_i} auswählen:

Beispiel 6.1. Wir betrachten eine Regel für die Zustandsgröße eines DRFS, für die die drei linguistischen Werte $L_1^x = \text{„klein“}$, $L_2^x = \text{„mittel“}$

und $L_3^x = \text{„groß“}$ mit den zugehörigen Kernpositionen s_1^x , s_2^x und s_3^x definiert sind. Ergibt sich nun als Auswahlvektor $\mathbf{a}_{w_i} = [0 \ 1 \ 0]$, so wird hierdurch gemäß Gl. (6.3)

$$s_{w(\mathbf{j}, \mathbf{q})}^x = \mathbf{a}_w \mathbf{s}^x = [0 \ 1 \ 0] \begin{bmatrix} s_1^x \\ s_2^x \\ s_3^x \end{bmatrix} = s_2^x$$

der Regel der linguistische Ausgangswert $L_2^x = \text{„mittel“}$ zugeordnet.

Mit Gl. (6.3) lässt sich Gl. (6.1) umschreiben zu

$$\begin{aligned} x_i(k+1) &= \sum_{\mathbf{j}, \mathbf{q}} s_{w_i(\mathbf{j}, \mathbf{q})}^{x_i} \Xi_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}, \mathbf{u}) = \sum_{\mathbf{j}, \mathbf{q}} \mathbf{a}_{w_i}^T \mathbf{s}^{x_i} \Xi_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}, \mathbf{u}) \\ &= \sum_{\mathbf{j}, \mathbf{q}} \mathbf{a}_{w_i}^T \mathbf{\Lambda}_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}, \mathbf{u}), \end{aligned} \quad (6.4)$$

wobei

$$\mathbf{\Lambda}_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}, \mathbf{u}) = \mathbf{s}^{x_i} \Xi_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}, \mathbf{u}).$$

Ersetzt man nun die Summe in Gl. (6.4) durch das Skalarprodukt, so erhält man

$$x_i(k+1) = \sum_{\mathbf{j}, \mathbf{q}} \mathbf{a}_{w_i}^T \mathbf{\Lambda}_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}, \mathbf{u}) = \mathbf{v}_i^T \mathbf{\Lambda}(\mathbf{x}, \mathbf{u}) = \mathbf{\Lambda}^T(\mathbf{x}, \mathbf{u}) \mathbf{v}_i, \quad (6.5)$$

wobei

$$\mathbf{v}_i = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_R \end{bmatrix} \in \{0,1\}^{Rr_x(i) \times 1}, \quad \mathbf{\Lambda}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{\Lambda}_1(\mathbf{x}, \mathbf{u}) \\ \vdots \\ \mathbf{\Lambda}_R(\mathbf{x}, \mathbf{u}) \end{bmatrix} \in \mathbb{R}^{Rr_x(i) \times 1} \quad (6.6)$$

gilt und R die Zahl der Regeln angibt.

Mit dieser Herleitung lässt sich nun das Optimierungsproblem aufstellen. Wie in Modellierungsproblem 6.1 angegeben, soll der quadratische Fehler minimiert werden, so dass mit Gl. (6.5) das Gütekriterium

$$\begin{aligned} J_i(\mathbf{v}_i) &= \frac{1}{2} \sum_{v=1}^N (x_i^{(v)}(k+1) - \hat{x}_i^{(v)}(k+1))^2 \\ &= \frac{1}{2} \sum_{v=1}^N (x_i^{(v)}(k+1) - \mathbf{\Lambda}^T(\mathbf{x}^{(v)}(k), \mathbf{u}^{(v)}(k)) \mathbf{v}_i)^2 \end{aligned} \quad (6.7)$$

folgt. Definiert man nun wiederum

$$\mathbf{z}_i = \begin{bmatrix} x_i^{(1)}(k+1) \\ \vdots \\ x_i^{(N)}(k+1) \end{bmatrix} \in \mathbb{R}^{N \times 1}, \quad \mathbf{\Upsilon} = \begin{bmatrix} \mathbf{\Lambda}^T(\mathbf{x}^{(1)}(k), \mathbf{u}^{(1)}(k)) \\ \vdots \\ \mathbf{\Lambda}^T(\mathbf{x}^{(N)}(k), \mathbf{u}^{(N)}(k)) \end{bmatrix} \in \mathbb{R}^{N \times Rr_x(i)},$$

so kann man das Gütekriterium (6.7) kompakter durch

$$\begin{aligned} J_i(\mathbf{v}_i) &= \frac{1}{2} (\mathbf{z}_i - \mathbf{\Upsilon} \mathbf{v}_i)^T (\mathbf{z}_i - \mathbf{\Upsilon} \mathbf{v}_i) \\ &= \frac{1}{2} \mathbf{v}_i^T \mathbf{\Upsilon}^T \mathbf{\Upsilon} \mathbf{v}_i - \mathbf{z}_i^T \mathbf{\Upsilon} \mathbf{v}_i + \frac{1}{2} \mathbf{z}_i^T \mathbf{z}_i \end{aligned} \quad (6.8)$$

darstellen. Die zu optimierenden Parameter \mathbf{v}_i sind hierbei binäre Variablen. Mit dem Gütekriterium (6.8) ist das Problem allerdings noch nicht vollständig definiert, da die binären Optimierungsparameter bisher noch beliebige Werte null und eins annehmen können. Dies ist offensichtlich nicht sinnvoll, da die Parameter Auswahlvektoren \mathbf{a}_{w_i} definieren. Da einer Regelprämisse maximal eine Ausgangskernposition zugeordnet werden kann, müssen Beschränkungen an die Parameter gestellt werden. Hierzu gibt es prinzipiell zwei Möglichkeiten, die unterschiedlich begründet werden können und im Folgenden beschrieben werden.

6.2.1 Ganzzahlige Optimierung mit Nebenbedingungen

Wie bereits angesprochen, setzt sich der Vektor der binären Optimierungsparameter \mathbf{v}_i nach Gl. (6.6) aus den Auswahlvektoren \mathbf{a}_{w_i} zusammen. Da jeder Regelprämisse eine Ausgangskernposition zugeordnet werden sollte, liegt es nahe, für die Einträge eines Auswahlvektors \mathbf{a}_{w_i}

$$\sum_{k=1}^{r(i)} a_k = 1 \quad (6.9)$$

zu fordern. Da die Komponenten binäre Variablen sind, bedeutet dies, dass genau ein Eintrag gleich eins sein sollte. Die zugehörige Kernposition im Vektor \mathbf{s}^{x_i} ist dann die Ausgangskernposition für die betrachtete Regel. Da insgesamt R Regeln vorhanden sind, entstehen R Nebenbedingungen der Form (6.9), die sich in Vektorschreibweise zu

$$\mathbf{A}_{nb} \mathbf{v}_i = \mathbf{b}_{nb} \quad (6.10)$$

mit

$$\mathbf{A}_{nb} = \begin{bmatrix} \mathbf{1}_{1 \times r_x(i)} & \mathbf{0}_{1 \times r_x(i)} & \cdots & \mathbf{0}_{1 \times r_x(i)} \\ \mathbf{0}_{1 \times r_x(i)} & \mathbf{1}_{1 \times r_x(i)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0}_{1 \times r_x(i)} \\ \mathbf{0}_{1 \times r_x(i)} & \cdots & \mathbf{0}_{1 \times r_x(i)} & \mathbf{1}_{1 \times r_x(i)} \end{bmatrix}, \quad \mathbf{b}_{nb} = \mathbf{1}_{Rr_x(i) \times 1}$$

zusammenfassen lassen. Hierbei sind $\mathbf{1}_{m \times n}$ bzw. $\mathbf{0}_{m \times n}$ Matrizen der Dimension $m \times n$, deren Einträge ausschließlich eins bzw. null sind. Durch Kombination der Gütefunktion (6.8) mit den Nebenbedingungen (6.10) ergibt sich schließlich:

Optimierungsproblem 6.1.

$$\begin{aligned} \min_{\mathbf{v}_i} \quad & \frac{1}{2} \mathbf{v}_i^T \mathbf{\Upsilon}^T \mathbf{\Upsilon} \mathbf{v}_i - \mathbf{z}_i^T \mathbf{\Upsilon} \mathbf{v}_i \\ \text{sodass} \quad & \mathbf{A}_{nb} \mathbf{v}_i = \mathbf{b}_{nb}, \\ & \mathbf{v}_i \in \{0,1\}^{Rr_x(i)}. \end{aligned}$$

In der Herleitung von Optimierungsproblem 6.1 wurde vorausgesetzt, dass einer Regelprämisse *genau eine* Ausgangskernposition zugeordnet werden muss. Dies ist nicht immer sinnvoll, da die Datensätze mitunter nur einen Teil des Zustandsraums abdecken. Dies kann zum einen auftreten, wenn die Datensätze nur bestimmte, für die Anwendung interessante Betriebsbereiche abdecken. Zum anderen kann das System auch aus physikalischen Gründen Bereiche des Zustandsraums nicht erreichen. Da sich das System in einem solchen Bereich nicht aufhalten kann, kann auch keine sinnvolle Regel definiert werden. In solchen Fällen sollte also der entsprechenden Regel *keine* Ausgangskernposition zugewiesen werden. Um undefinierte Konklusionen in der Regelbasis zu vermeiden, kann die Ausgangskernposition einer solchen Regel nach der Optimierung als Interpolation der benachbarten Kernpositionen gemäß Gl. (5.11) in die Regelbasis übernommen werden. Auf diese Weise ist das DRFS auch dann definiert, wenn das System in einem solchen unbekannten Betriebspunkt betrieben wird. Eine solche Bedingung kann direkt im Optimierungsproblem berücksichtigt werden, indem die Nebenbedingung (6.10) durch eine Ungleichungsnebenbedingung ersetzt wird. Dann ergibt sich:

Optimierungsproblem 6.2.

$$\begin{aligned}
& \min_{\mathbf{v}_i} \quad \frac{1}{2} \mathbf{v}_i^T \mathbf{\Upsilon}^T \mathbf{\Upsilon} \mathbf{v}_i - \mathbf{z}_i^T \mathbf{\Upsilon} \mathbf{v}_i \\
& \text{sodass} \quad \mathbf{A}_{nb} \mathbf{v}_i \leq \mathbf{b}_{nb}, \\
& \quad \mathbf{v}_i \in \{0,1\}^{Rr_x(i)}.
\end{aligned}$$

Sowohl bei Optimierungsproblem 6.1 als auch bei Optimierungsproblem 6.2 handelt es sich um ganzzahlige quadratische Optimierungsprobleme mit linearen Nebenbedingungen. Hieraus folgt direkt, dass das relaxierte Problem ein quadratisches, und damit konvexes Optimierungsproblem ist. Somit lassen sich Branch-and-Bound Verfahren zur Lösung verwenden. Diese bilden auch die Grundlage für eine Reihe effektiver Solver für solche Probleme, wie beispielsweise CPLEX [76], BARON [164] oder der in dieser Arbeit verwendete, frei erhältliche Solver MIQP [23]. Wichtig zu erwähnen ist die Tatsache, dass für das ganzzahlige quadratische Optimierungsproblem und damit für Teilproblem 1, also die Identifikation der Regelbasis, aufgrund der Konvexität der relaxierten Probleme in jedem Fall das globale Optimum gefunden wird.

Ein Nachteil des vorgestellten Ansatzes besteht darin, dass die Zahl der ganzzahligen Optimierungsparameter wiederum exponentiell mit der Zahl der Regeln ansteigt, wodurch sich numerische Probleme bei den Solvern bei der Optimierung von Systemen hoher Dimension ergeben können. Auf dieses bei RFS inhärente Problem wurde bereits im vorigen Kapitel eingegangen. Mindern lässt sich dieses Problem zum einen durch die Reduktion von Parametern, beispielsweise mithilfe der Auswahl relevanter Eingangs- und Zustandsgrößen nach Abschnitt 5.4, die auch auf DRFS mit Rückkopplungskorrespondenz anwendbar ist. Zum anderen reicht zumeist eine vergleichsweise niedrige Zahl an Zugehörigkeitsfunktionen pro Dimension (z.b. 2-4) aus, da das DRFS hauptsächlich die qualitativen Zusammenhänge des Modells wiedergeben soll. Dadurch ist die Zahl der Regeln bei einem qualitativen Modell im Allgemeinen geringer als bei einem genauen Approximationsmodell.

6.3 Parameteroptimierung

Zur Optimierung der Parameter, also der Eingangskernpositionen \mathbf{s}_j^x , \mathbf{s}_q^u und der Ausgangskernpositionen \mathbf{s}_w^x , wird, wie bereits bei zeitkontinuierlichen RFS, ein gradientenbasiertes Verfahren entwickelt. Prinzipiell ist das

Optimierungsproblem sowie dessen Lösung der im Abschnitt 5.2 vorgestellten Optimierung KRFS sehr ähnlich. Aufgrund der speziellen Struktur, die durch die Rückkopplungskorrespondenz entsteht, ergeben sich allerdings einige Veränderungen, auf die im Folgenden näher eingegangen wird. Im Gegensatz zur Optimierung der KRFS ist hier zu beachten, dass die Ausgangskernpositionen aufgrund der Rückkopplungskorrespondenz aus der Menge der Eingangskernpositionen stammen.

Dies hat Folgen für das Optimierungsproblem. Im Gegensatz zur Optimierung von KRFS kann die Optimierung der Ausgangs- und Eingangszugehörigkeitsfunktionen nicht mehr getrennt betrachtet werden. Somit ist die Optimierung der Ausgangszugehörigkeitsfunktionen nicht mehr als Least-Squares-Problem darstellbar und nicht konvex. Vielmehr ergibt sich, wie schon für die Kernpositionen im Zeitkontinuierlichen, ein nichtkonvexes, multimodales Optimierungsproblem.

Da die Lage der Ausgangskernpositionen aufgrund der Rückkopplungskorrespondenz mit der Lage der Eingangskernpositionen identisch ist, verringert sich die Zahl der effektiv zu optimierenden Parametern auf die Eingangskernpositionen

$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{s}_j^x \\ \mathbf{s}_q^u \end{bmatrix}, \quad \boldsymbol{\xi} \in \mathbb{R}^{l \times 1}, \quad l = \sum_{\alpha=1}^{n+m} r(\alpha).$$

Diese Parameter werden nun mithilfe des Gradientenverfahrens iterativ gemäß

$$\boldsymbol{\xi}(\kappa) = \boldsymbol{\xi}(\kappa - 1) - \eta(\kappa - 1)\mathbf{G}(\kappa - 1)$$

optimiert, wobei κ den Iterationsschritt, η die Schrittweite und \mathbf{G} den Gradienten der Gütefunktion bezeichnet. Die Gütefunktion ist wieder der quadratische Fehler zwischen gemessenem $\mathbf{x}_i(k+1)$ und dem anhand von $\mathbf{x}^{(v)}(k)$, $\mathbf{u}^{(v)}(k)$ berechneten Ausgangswert $\hat{\mathbf{x}}_i(k+1)$:

$$J_i(\boldsymbol{\xi}) = \frac{1}{2} \sum_{v=1}^N \left(e_i^{(v)} \right)^2 = \frac{1}{2} \sum_{v=1}^N \left(x_i^{(v)}(k+1) - \hat{x}_i^{(v)}(k+1) \right)^2.$$

Wie im Zeitkontinuierlichen weist der Index i darauf hin, dass insgesamt n Gütefunktionen J_i auftreten, die jeweils für eine Komponente des Zustands x_i gelten. Entsprechend entsteht wieder ein Mehrzieloptimierungsproblem der Form (5.16), das gemäß Optimierungsproblem (5.17) mithilfe der gewichteten Summe in ein einzelnes Optimierungsproblem überführt wird.

Abschließend bestimmt sich der Gradient der Gütefunktion (5.17) zu

$$\mathbf{G} = \frac{\partial J(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = \frac{\partial (\boldsymbol{\tau}^T \cdot \mathbf{J}(\boldsymbol{\xi}))}{\partial \boldsymbol{\xi}} = \boldsymbol{\tau}^T \cdot \frac{\partial \mathbf{J}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}},$$

mit der Jacobi-Matrix (5.19). Deren Einträge ergeben sich allgemein zu

$$\frac{\partial J_i(\boldsymbol{\xi})}{\partial \xi_\lambda} = - \sum_{v=1}^N \left(x_i^{(v)} - \hat{x}_i^{(v)} \right) \cdot \frac{\partial \hat{x}_i^{(v)}(\boldsymbol{\xi})}{\partial \xi_\lambda},$$

wobei sich bei DRFS

$$\frac{\partial \hat{x}_i^{(v)}(\boldsymbol{\xi})}{\partial \xi_\lambda} = \sum_{\mathbf{j}, \mathbf{q}} \left[\frac{s_{w_i(\mathbf{j}, \mathbf{q})}^{x_i}}{\partial \xi_\lambda} \Xi_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}^{(v)}, \mathbf{u}^{(v)}) + s_{w_i(\mathbf{j}, \mathbf{q})}^{x_i} \frac{\partial \Xi_{(\mathbf{j}, \mathbf{q})}(\mathbf{x}^{(v)}, \mathbf{u}^{(v)})}{\partial \xi_\lambda} \right] \quad (6.11)$$

ergibt. Gl. (6.11) resultiert aus der Anwendung der Produktregel und setzt sich entsprechend aus zwei Ausdrücken zusammen. Der zweite Ausdruck in der Klammer ist bereits aus Abschnitt 5.2.2 bekannt und berechnet sich gemäß den Gl. (5.20) und Gl. (5.21). Der erste Ausdruck entsteht durch die Ausgangskernpositionen, die gemäß der Zuordnung in der Regelbasis Eingangskernpositionen entsprechen. Für deren Ableitung gilt nun

$$\frac{s_{w_i(\mathbf{j}, \mathbf{q})}^{x_i}}{\partial \xi_\lambda} = \begin{cases} 1 & \text{wenn } \xi_\lambda = s_{w_i(\mathbf{j}, \mathbf{q})}^{x_i} \\ 0 & \text{sonst.} \end{cases}$$

Für die Ableitungen nach den Kernpositionen \mathbf{s}_q^u fällt der erste Ausdruck in Gl. (6.11) weg. Für den zweiten Ausdruck ergeben sich Ergebnisse entsprechend Gl. (5.20) und Gl. (5.21).

6.4 Beispiel: Thermofluidischer Prozess

Die in den vorigen Abschnitten vorgestellten Verfahren zur qualitativen und quantitativen Modellierung mit DRFS sollen nun an einem thermofluidischen Prozess, wie ihn Abbildung 6.1 zeigt, verifiziert werden. Die Systembeschreibung ist [66] entnommen. Der Prozess besteht aus einem zylindrischen Batch-Reaktor, der von einem Tank und einem zusätzlichen Wasserzufluss gespeist wird. Die Zuflüsse können jeweils kontinuierlich über Ventile eingestellt werden. Der Abfluss ist dagegen nur vom momentanen Füllstand abhängig. Des Weiteren kann die Temperatur des Fluids im Batch-Reaktor mithilfe von Heizstäben kontinuierlich erhöht werden.

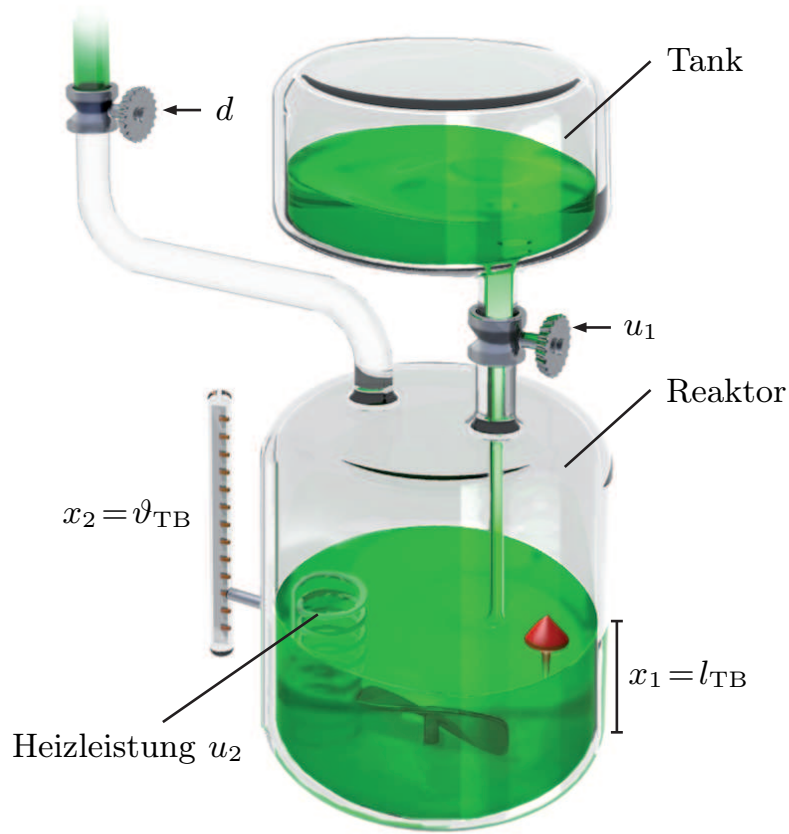


Abbildung 6.1: Thermofluidischer Prozess.

Als Stellgrößen stehen die Öffnung $u_1 \in [0, 1]$ des Einlassventils V_1 sowie die Heizleistung der Heizstäbe $u_2 \in [0, 6]$ zur Verfügung. Der Wasserzufluss, der sich aus der Öffnung von Ventil V_2 ergibt, wird als Störgröße d modelliert. Aufgrund des geringen Einflusses auf die Dynamik wird sie im Folgenden vernachlässigt. Durch Wahl der Zustandsgrößen Füllstand und Reaktortemperatur, also $\mathbf{x}(k) = (l_{TB}(k), \vartheta_{TB}(k))^T = (x_1(k), x_2(k))^T$ ergibt sich:

$$\begin{aligned} \dot{x}_1(t) &= \frac{1}{A_h} \left(q_{T_3}(u_1(t)) - K_A \sqrt{2gx_1(t)} \right), \\ \dot{x}_2(t) &= \frac{1}{V(x_1(t))} (q_{T_3}(u_1(t))(\vartheta_U - x_2(t)) + k_1 u_2(t)), \end{aligned} \quad (6.12)$$

wobei

$$q_{T_3}(u_1(t)) = \begin{cases} 7 \cdot 10^{-6} (11,1 u_1^2(t) + 13,1 u_1(t) + 0,2), & \text{wenn } u_1(t) > 0,2; \\ 0 & \text{sonst} \end{cases}$$

und

$$V(x_1(t)) = 0,07x_1(t) - 1,9 \cdot 10^{-3}, \quad k_1 = \frac{k_h P_{el}}{\rho c_p}.$$

Die betrachteten Volumenströme q_i werden in m^3/s , das Volumen V in m^3 , die Füllhöhe im Tank x_1 in m und die Temperatur x_2 in K . Die weiteren Parameter sind im Anhang in Tabelle B.3 angegeben. Der Zustandsraum $\mathcal{X} = [0,26, 0,45]\text{m} \times [273,15, 313,15]\text{K}$ wird als Arbeitsbereich festgelegt. Der Trainingsdatensatz, bestehend aus 135 Datenpaaren, wird durch Anregung des Systems mit APRB-Signalen an verschiedenen Arbeitspunkten erzeugt (Abbildungen B.4 und B.5 im Anhang). Hierdurch wird erreicht, dass der gesamte Bereich \mathcal{X} ausreichend durch Daten abgedeckt ist. Der Validierungsdatensatz (45 Datenpaare) wird in ähnlicher Weise erzeugt. Wie bereits in der Einführung angesprochen, wird eine relativ große Abtastzeit von $T_0 = 100\text{s}$ gewählt.

Mithilfe der so generierten Datensätze wird das System nun mit einem DRFS nachgebildet. Hierzu werden je drei Zugehörigkeitsfunktionen für die Zustände und je zwei Zugehörigkeitsfunktionen für die Eingänge gewählt. Die initialen Kernpositionen werden dabei äquidistant über den Wertebereich der jeweiligen Variablen verteilt. Des Weiteren kann mithilfe von Vorwissen bzw. durch Anwendung des in Abschnitt 5.4 beschriebenen Verfahrens zur Variablenauswahl die Zahl an relevanten Zuständen und Eingängen für die erste Zustandsgleichung auf den Zustand x_1 und den Eingang u_1 reduziert werden. Damit sind für Zustand x_1 sechs Regeln und für Zustand x_2 36 Regeln mithilfe der Optimierung festzulegen. Diese qualitative Modellierung geschieht nun durch Lösung von Optimierungsproblem 6.1, das für beide Zustände unabhängig voneinander gelöst werden kann. Das Ergebnis ist in Form eines linguistischen Automaten in Abbildung 6.2 dargestellt. Anhand des erhaltenen qualitativen Modells lässt sich die Systemdynamik bereits sehr gut beschreiben. Hierzu wird der linguistische Automat zunächst bezüglich der ersten Zustandsgröße x_1 untersucht. Es ergeben sich die folgenden sechs Regeln, die offenbar in gutem Einklang

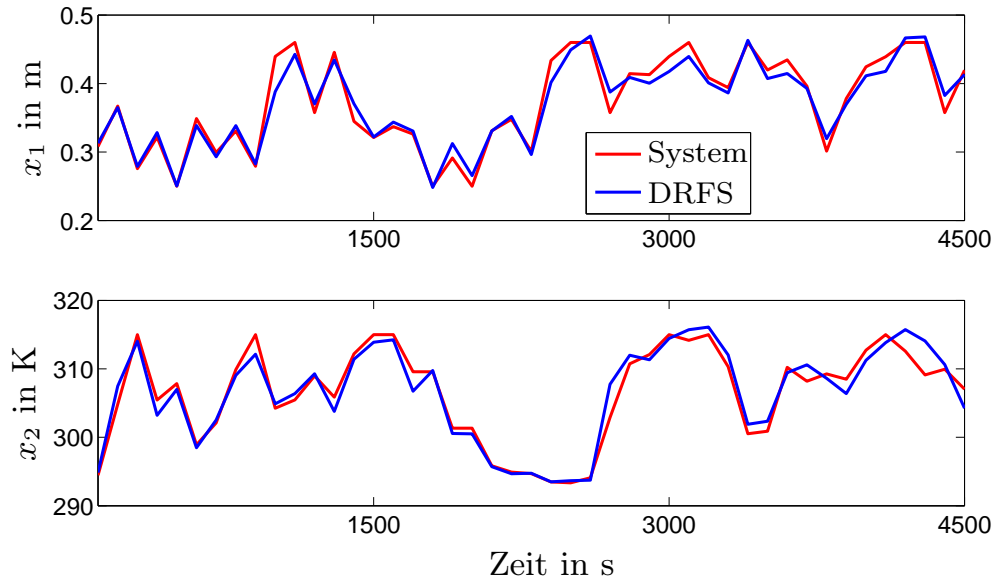


Abbildung 6.3: Vergleich der Verläufe der Zustände für den Validierungsdatensatz bei einer Abtastzeit von $T_I = 100\text{s}$ (Einschrittprädiktion). Das DRFS kann die qualitative Dynamik des Systems auch für unbekannte Daten gut nachbilden.

Es ergibt sich aber auch hier ein nachvollziehbares qualitatives Ergebnis. So ist beispielsweise anhand des linguistischen Automaten ablesbar, dass sich die Temperatur auch bei niedrigen Initialwerten bei Anregung nach einiger Zeit in den Bereich zwischen „mittel“ und „groß“ bewegt und unabhängig von der Anregung dort verbleibt. Des Weiteren steigt die Temperatur bei hoher Heizleistung erwartungsgemäß an. Bei niedriger Heizleistung und mittlerem Füllstand bleibt die Temperatur weitestgehend konstant. Bei hoher Temperatur und großem Zufluss dagegen nimmt die Temperatur ab. Dieses Verhalten entspricht der Systembeschreibung (6.12), da bei Temperaturen x_2 oberhalb der Umgebungstemperatur ϑ_U der Zufluss die Temperatur verringert.

Abschließend werden die Kernpositionen des erhaltenen qualitativen Modells mithilfe der Parameteroptimierung aus Abschnitt 6.3 nachoptimiert. Abbildung 6.3 zeigt die resultierenden Verläufe des Validierungsdatensatzes für beide Zustandsgrößen. Man erkennt, dass das DRFS den Systemverlauf gut nachbilden kann. Es ergeben sich allerdings an einigen Stellen auch Abweichungen vom realen Prozess. Diese sind durch die niedrige Zahl von nur zehn freien Parametern zu erklären. Zur hochgenauen Approximation des Systemverhaltens weist das DRFS zu wenige Freiheitsgrade auf. Trotzdem kann das erhaltene DRFS auch vergleichsweise gute quantitative Informationen über das System liefern, was durch andere qualitative Modellstrukturen, wie beispielsweise Automaten, nicht möglich ist.

6.5 Zusammenfassung und Fazit

Dieses Kapitel befasste sich mit der qualitativen Modellierung dynamischer Systeme mithilfe zeitdiskreter RFS. Das Problem wurde in die beiden Teilprobleme Identifikation der Regelbasis und Optimierung der freien Parameter zerlegt. Das erste Teilproblem lässt sich in ein ganzzahliges quadratisches Optimierungsproblem überführen, dessen Lösung das globale Optimum darstellt. Das zweite Teilproblem ist ein nichtkonvexes kontinuierliches Optimierungsproblem, für dessen Lösung ein Gradientenverfahren angegeben wird. Die Ergebnisse zeigen eine sehr gute *qualitative* Nachbildung dynamischer Systeme. Die Fähigkeiten DRFS zur quantitativ genauen Nachbildung sind durch die geringe Zahl an Optimierungsparametern zwar eingeschränkt. Im Gegensatz zu anderen qualitativen Modellstrukturen erreichen DRFS aber immer noch eine gute Approximationsgenauigkeit. Insgesamt eignen sich obige Modelle insbesondere für die Prozessüberwachung, bei der abstraktere, weniger genaue Modelle benötigt werden.

Abschließend sei noch darauf hingewiesen, dass die Verfahren zur Identifikation der Regelbasis mit ganzzahliger quadratischer Optimierung nicht nur zur qualitativen Modellierung sondern auch zur Auslegung und Optimierung von DRFS zur Mustererkennung und Fehlerisolation eingesetzt werden können. Hierauf wird im folgenden Kapitel bei der Anwendung DRFS zur Fehlerisolation nochmals eingegangen.

7 Fehlerdiagnose in dynamischen Systemen

Dieses Kapitel beschäftigt sich mit der Verwendung rekurrenter Fuzzy-Systeme zur Erkennung und Isolation von Fehlern in technischen Systemen. Zunächst wird die Eignung des RFS zur Fehlerdetektion betrachtet und gezeigt, wie geeignete Residuen aus trainierten RFS gewonnen werden können. Anschließend wird dann die Verwendung des RFS zur Fehlerisolation gezeigt. Dieser Ansatz erweitert bisherige Ansätze zur Fehlerisolation mittels statischer Fuzzy-Systeme und eignet sich insbesondere zur Isolation von dynamischen Fehlern sowie zur Mehrfehlerisolation.

7.1 Einführung

Mit steigenden Anforderungen an die Effizienz, Produktqualität und Sicherheit von industriellen Prozessen wird eine zuverlässige Überwachung und Fehlerdiagnose zunehmend wichtiger. Diagnoseverfahren werden vermehrt eingesetzt, um frühzeitig Fehler zu erkennen und dadurch größere Schäden an der Anlage sowie unnötige Kosten durch eingeschränkten Betrieb bzw. Anlagenstillstand zu vermeiden.

Die Fehlerdiagnose lässt sich allgemein in drei Teilgebiete aufteilen: Fehlerdetektion, Fehlerisolation und Fehleridentifikation. Die Detektion dient der Erkennung von Abweichungen bestimmter Prozessgrößen vom normalen Betriebszustand und stellt somit fest, dass ein Fehler in der Anlage aufgetreten ist. Die nachfolgende Fehlerisolation bestimmt dann, welcher der möglichen Fehlerfälle tatsächlich eingetreten ist. Optional kann mithilfe der Fehleridentifikation die Größe oder das Ausmaß der aufgetretenen Fehler bestimmt werden. Prinzipiell lässt sich das Vorgehen zur Fehlerdetektion und Isolation (FDI) gemäß Abbildung 7.1 in drei Schritte unterteilen. Ausgangspunkt ist der Prozess, in dem Fehler auftreten können. Mithilfe der Messdaten aus dem Prozess \mathbf{x}_{mess} können dann sogenannte Symptome generiert werden, die anschließend mit Nominalwerten verglichen werden. Diese Vergleichsgrößen werden als Residuen \mathbf{r} bezeichnet. Durch Auswer-



Abbildung 7.1: Teilschritte der Fehlerdiagnose.

tung der Residuen können dann die Fehler φ detektiert und isoliert werden.

Methoden zur Generierung von geeigneten Symptomen bzw. Residuen lassen sich grob in signalgestützte und modellgestützte Verfahren unterteilen [80]. Zu den signalgestützten Verfahren gehören die Grenzwert- und Trendüberwachung sowie verschiedene Methoden zur Signalanalyse wie Korrelation oder Spektralanalysen. In den letzten Jahren werden zunehmend modellbasierte Verfahren eingesetzt. Hierbei lassen sich beobachterbasierte Verfahren [53], Verfahren zur Parameterschätzung [77] und Verfahren auf Basis von Paritätsgleichungen [60] unterscheiden. Die beobachterbasierten Ansätze beruhen auf einem um die Fehlerdynamik erweiterten Zustandsbeobachter. Dieser wird so ausgelegt, dass bei Auftreten eines Fehlers mindestens ein Zustand abweicht, wodurch der Fehler detektiert werden kann. Bei der Parameterschätzung werden charakteristische Größen des Systems, beispielsweise Widerstände oder Durchflusskoeffizienten online geschätzt. Weichen diese Schätzwerte vom Nominalwert ab, wird auf einen Fehler geschlossen. Paritätsgleichungen sind den Prozess beschreibende Gleichungen, die messbare Ausgangs- und Eingangsgrößen miteinander verknüpfen. Ergeben sich Differenzen zwischen den beiden Seiten der Gleichung, so ist offenbar ein Fehler aufgetreten.

Die obigen Methoden basieren jeweils auf analytischen Zusammenhängen, die im Regelfall mithilfe einer physikalischen Modellbildung gebildet wurden. Dies ist, wie bereits beschrieben, bei einigen Prozessen nicht möglich. In solchen Fällen können Verfahren aus der künstlichen Intelligenz, wie neuronale Netze oder Fuzzy-Systeme, als Modellbildungswerkzeuge verwendet werden [54, 55, 152]. So werden in [150] Beobachter auf Basis von TSK Fuzzy-Systemen zur Fehlerdiagnose in dynamischen Systemen entworfen. Des Weiteren wurde das bereits vorgestellte Neuro-Fuzzy-System *Lolimot* zur modellbasierten Fehlerdiagnose eingesetzt [16]. Ähnliche Methoden finden sich u.a. in [122, 218]. Basierend auf den in den vorigen Kapiteln eingeführten Verfahren wird in diesem Abschnitt eine modellba-

sierte Residuengenerierung vorgestellt [175], die sich von obigen Ansätzen dahingehend unterscheidet, dass sie RFS zur Modellierung nutzt.

Außerdem können Fuzzy-Systeme bei der Auswertung der Residuen eingesetzt werden. So schlagen [167, 168] adaptive Schwellen basierend auf Fuzzy-Logik für die Fehlerdetektion vor. Auch zur Fehlerisolation existieren verschiedene Ansätze. Prinzipiell ist die Fehlerisolation der Mustererkennung sehr ähnlich. Entsprechend können verschiedene Arten von Klassifikatoren [41, 113] oder Fuzzy-Fehlerbäume [57, 104] eingesetzt werden. Eine weitere Möglichkeit sind Inferenzmethoden, die zumeist auf der Auswertung von Fehlersymptomtabellen basieren [61]. Hierzu werden vielfach statische Fuzzy-Systeme verwendet [80, 103, 208]. Andere Verfahren basieren auf Neuro-Fuzzy-Systemen [11, 156]. Die Anwendung von dynamischen Fuzzy-Systemen, wie die in dieser Arbeit betrachteten RFS, zur Fehlerisolation auf Basis der Auswertung von Fehlersymptomtabellen existiert dagegen bisher nicht.

Über die zuvor vorgestellten Methoden hinaus gibt es noch weitere Verfahren wie die auf ereignisdiskreten Systemen beruhenden konsistenzbasierte Fehlerdiagnose [36, 126, 165, 166, 170, 185]. Hierbei wird das Prozessmodell soweit abstrahiert, dass nurmehr die qualitativen Zusammenhänge beschrieben werden. Typische Beispiele solcher Prozessmodelle sind Petri-Netze oder Automaten [127]. Werden nun für die einzelnen Fehlerfälle abstrahierte Modelle gebildet, so können die am Prozess gemessenen Daten auf Konsistenz mit diesen Modellen untersucht werden. Es wird also geprüft, ob das Modell die gemessene Signalfolge generieren kann. Ist dies nicht der Fall, so kann der diesem Modell zugrunde liegende Fehler ausgeschlossen werden. So werden nach und nach nichtkonsistente Fehlerfälle ausgeschlossen. Eine Übersicht über die verschiedenen Verfahren zur FDI findet sich u.a. in [25, 80, 106, 198, 199, 200].

Im Folgenden soll beleuchtet werden, bei welchen Teilschritten der FDI RFS eingesetzt werden können [147, 175]. Abschnitt 7.2 zeigt, wie mithilfe einer auf RFS basierenden Modellbeschreibung geeignete Residuen generiert werden können. Abschnitt 7.3 beschäftigt sich dann mit der Fehlerisolation. Hier wird ein Ansatz vorgestellt, der mithilfe DRFS mit Rückkopplungskorrespondenz insbesondere die Isolation von Fehlern mit dynamischen Eigenschaften sowie die Isolation mehrerer nacheinander auftretender Fehler ermöglicht.

7.2 Residuengenerierung

Von zentraler Bedeutung für die modellgestützte Fehlerdetektion ist die Generierung geeigneter Residuen, die den Status des Prozesses beschreiben und somit Abweichungen vom fehlerfreien Systemzustand anzeigen. Hierzu benötigt man ein genaues Modell des Systems, das oftmals mithilfe einer physikalischen Modellbildung bestimmt werden kann. Allerdings hat man es des Öfteren auch mit Prozessen zu tun, bei denen eine physikalische Modellbildung nicht oder kaum möglich ist. Dann müssen datengestützte Verfahren zur Modellbildung verwendet werden.

Zur Modellbildung mit RFS können je nach vorhandenen Messdaten und Vorwissen die in den Kapiteln 5 und 6 beschriebenen Verfahren verwendet werden. Hierbei können abhängig von den im Betrieb vorhandenen Messdaten entweder nur die Dynamikfunktion oder auch zusätzlich die Ausgangsfunktion modelliert werden.

Es stellt sich nun die Frage, wie auf Basis eines solchen Modells geeignete Residuen gebildet werden können, mit denen sich Fehler detektieren lassen. Da DRFS nichtlineare Zustandsraummodelle des Systems darstellen, bieten sich als Residuen \mathbf{r} insbesondere die Differenzen zwischen gemessenen \mathbf{x} und durch das DRFS prädizierten Systemzuständen $\hat{\mathbf{x}}$,

$$\mathbf{r}_x = \mathbf{x} - \hat{\mathbf{x}}, \quad (7.1)$$

an. In diesem Fall müssen die Zustände des Systems auch während des Betriebs gemessen werden können. Stehen dagegen während des Betriebs nicht alle Zustände, sondern nur Ausgangsdaten zur Verfügung, so ergeben sich als Residuen die Differenzen zwischen gemessenen \mathbf{y} und durch das DRFS prädizierten Ausgängen $\hat{\mathbf{y}}$,

$$\mathbf{r}_y = \mathbf{y} - \hat{\mathbf{y}}.$$

Die Residuen können nun durch Parallelschaltung von DRFS und realem Prozess generiert werden. Tritt dann ein Fehler im Prozess auf, stimmen die Zustände x_i bzw. Ausgänge y_v des fehlerhaften Prozesses nicht mehr mit den Zuständen \hat{x}_i bzw. Ausgängen \hat{y}_v des DRFS überein und $r_{x,i}$ bzw. $r_{y,v}$ weicht von null ab. Abbildung 7.2 zeigt die Struktur.

In einigen Fällen führt diese Struktur aber dazu, dass sich bei Auftreten eines Fehlers im Allgemeinen alle Residuen $r_{x,i}$ bzw. $r_{y,v}$ verändern. Wünschenswert für eine gute Fehlerisolation ist dagegen, dass ein Residuum nur auf einige wenige Fehler reagiert, also nur auf wenige Fehler sensitiv ist. Dies erleichtert dann die Fehlerisolation.

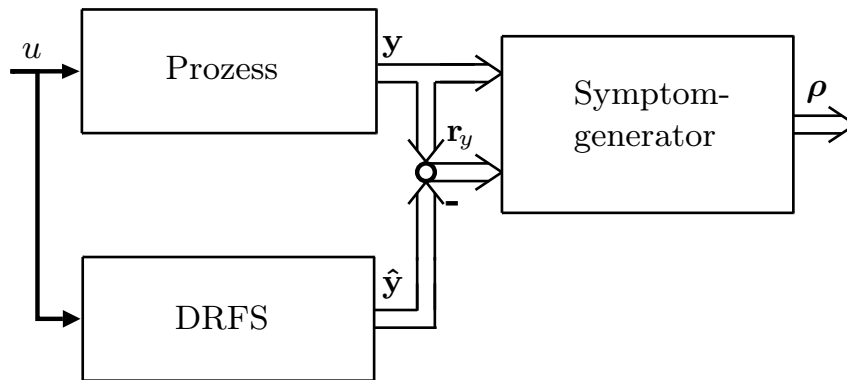


Abbildung 7.2: Parallelschaltung des realen Prozesses und eines rekurrenten Fuzzy-Systems zur Fehlerdetektion.

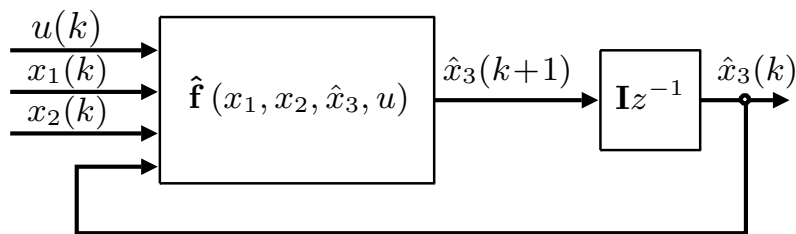


Abbildung 7.3: Veränderte Verschaltung eines zeitdiskreten rekurrenten Fuzzy-Systems mit drei Zuständen und einem Eingang zur Erhöhung der Fehlersensitivität der Residuen.

Falls alle Zustände messbar sind, kann die Fehlersensitivität der Residuen durch eine veränderte Verschaltung der DRFS, wie sie Abbildung 7.3 zeigt, erhöht werden. Im Gegensatz zur bekannten Struktur des DRFS (Abbildung 3.4) wird in dieser Anordnung nur noch eine Zustandsgröße \hat{x}_i zurückgeführt. Für die anderen Zustände \mathbf{x} werden die Messgrößen als zusätzliche Eingänge verwendet. Diese Anordnung hat den Vorteil, dass das so gebildete Residuum nur mehr sensitiv auf Fehler ist, die direkt auf die zurückgeführte Zustandsgröße wirken. Dagegen ist das Residuum durch die zusätzlich gemessenen restlichen Zustandsgrößen im Gegensatz zur reinen Parallelschaltung nicht sensitiv gegenüber Fehlern, die andere Zustandsgrößen beeinflussen.

Ein so aufgebautes DRFS wird nun für jede Zustandsgröße eingesetzt, wobei als Residuen wiederum die Differenzen der gemessenen und prädierten Zustände verwendet werden. Die resultierende Struktur zur Fehlerdetektion ist in Abbildung 7.4 gezeigt. Diese Residuen werden im Symptomgenerator mit den Residuen im fehlerfreien Fall \mathbf{r}_0 verglichen. Optional kann diese Größe auch noch umgerechnet werden, beispielsweise durch

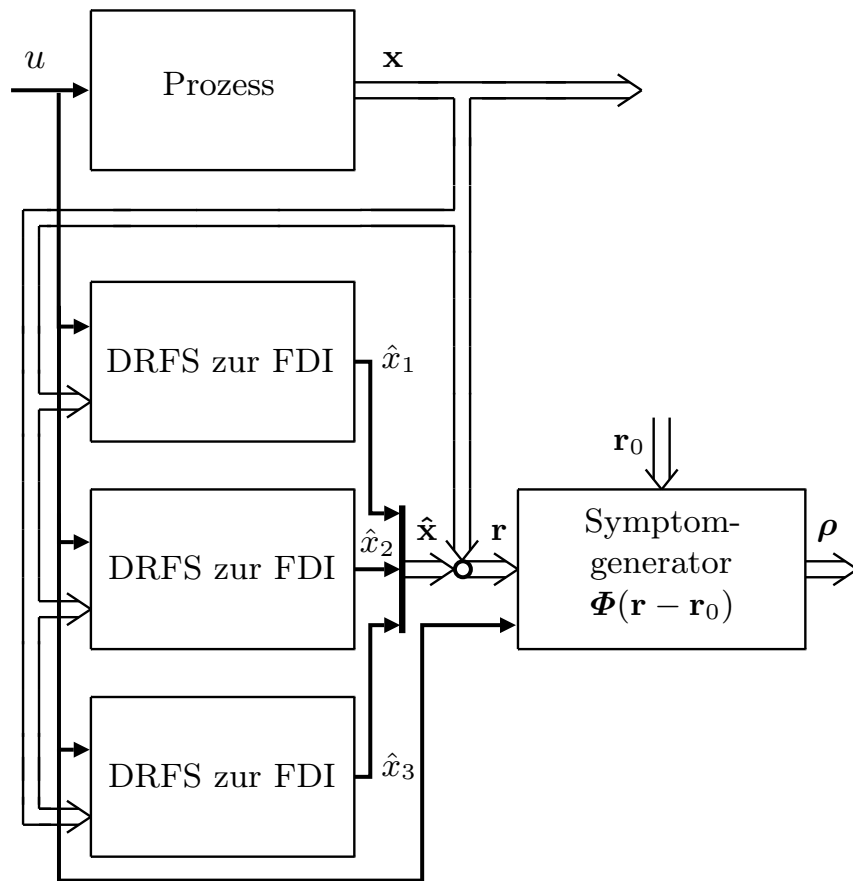


Abbildung 7.4: Struktur der Fehlerdetektion mit zeitdiskreten rekurrenten Fuzzy-Systemen für ein System mit drei Zuständen und einem Eingang.

eine Normierung auf den Eingangswert, um die Abhängigkeit vom Anregungssignal zu verringern. Als Ausgang ergibt sich der Fehlersymptomvektor als Funktion Φ der Residuen \mathbf{r} und \mathbf{r}_0 zu

$$\rho = \Phi(\mathbf{r}, \mathbf{r}_0) = \Phi(\mathbf{r} - \mathbf{r}_0).$$

Wichtig zu bemerken ist, dass immer dasselbe DRFS verwendet wird, so dass nur ein Modell des Prozesses datengestützt gebildet werden muss. Insbesondere benötigt der Ansatz keine Modelle des fehlerhaften Prozesses. Die DRFS unterscheiden sich lediglich in der Verschaltung, d.h. in der Zustandsgröße, die zurückgeführt wird. Auf Basis dieser Struktur werden nun modellgestützt Residuen und daraus Symptome generiert. Wie diese Symptome zur Fehlerisolation eingesetzt werden können, wird im nächsten Abschnitt erläutert.

7.3 Fehlerisolation

Viele Verfahren zur FDI konzentrieren sich auf die Diagnose von einzeln auftretenden Fehlern. Dies setzt voraus, dass nach Auftreten eines Fehlers dieser schnellstmöglich behoben wird, da die Fehlerdiagnose ansonsten keine verlässlichen Ergebnisse liefert. Eine schnelle Reparatur ist aber oftmals nicht möglich oder auch nicht erforderlich, insbesondere wenn die Fehler keinen kritischen Prozesszustand hervorrufen. In solchen Fällen können also bis zur Wartung mehrere Fehler auftreten, die detektiert und isoliert werden sollen.

Tatsächlich existieren in der Literatur nur wenige Ansätze zur Fehlerisolation, die den Mehrfehlerfall explizit mit betrachten. Erste Ansätze waren motiviert durch das Aufspüren von Fehlern in Bauteilen oder Verkabelung beim digitalen Schaltungsentwurf [38]. Die Methodik basiert auf einer qualitativen Modellierung der statischen Zusammenhänge. Erweiterungen auf dynamische Systeme basierend auf einer Abstraktion des Systems in Form von Automaten [182] oder Graphen [195, 197] wurden entwickelt. Weitere Ansätze verwenden sequentielle statistische Tests, um den wahrscheinlichsten (Mehr-) Fehlerfall zu ermitteln [144, 162, 178].

Allerdings berücksichtigen die obigen Ansätze keine Informationen über die zeitliche Entwicklung des Fehlerzustands. Es wird also zu einem bestimmten Zeitpunkt der Fehlerzustand ermittelt, ohne die vorherige Entwicklung der Fehler zu kennen und zu berücksichtigen. Entsprechend werden *gleichzeitig auftretende* Fehler isoliert. Der in dieser Arbeit verfolgte Ansatz geht dagegen von einer kontinuierlichen Überwachung des Prozesses aus, sodass Informationen über die Entwicklung der Fehler berücksichtigt werden können. Dies bietet Vorteile bei der Erkennung *mehrerer hintereinander auftretender* Fehler. Solche Fehler lassen sich prinzipiell auch mithilfe von Verfahren aus dem Bereich der ereignisdiskreten Systeme isolieren [34, 35, 134], allerdings wird hierzu wiederum ein abstrahiertes Systemmodell benötigt.

Das im Rahmen dieser Arbeit entwickelte Verfahren [175] arbeitet dagegen direkt mit den Symptomen. Damit ist der Ansatz mit jeder Form der Residuengenerierung (signalgestützt, modellbasiert oder konsistenzbasiert) kombinierbar. Zur Herleitung der Methodik wird im ersten Schritt ein statisches Fuzzy-System (SFS) zur Fehlerisolation vorgestellt. Dieses SFS stößt allerdings, wie sich zeigen wird, sowohl bei dynamischen Fehlerverläufen als auch im Mehrfehlerfall an seine Grenzen. Deshalb wird im zweiten Schritt eine Erweiterung auf DRFS vorgestellt, mit dem auch solche Fehler isoliert werden können.

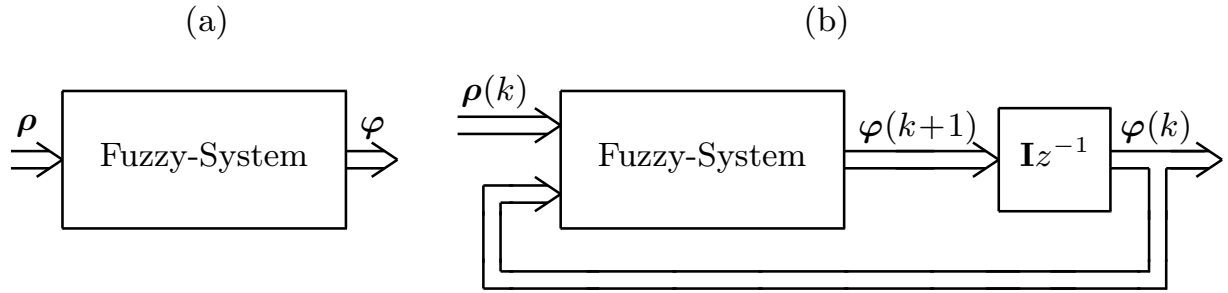


Abbildung 7.5: (a) Statisches Fuzzy-System zur Fehlerisolation. (b) Rekurrentes Fuzzy-System zur Fehlerisolation. Im Vergleich zum SFS wird der aktuelle Fehlerzustand zurückgeführt und zur Ermittlung des neuen Fehlerzustands verwendet.

7.3.1 Fehlerisolation mit statischen Fuzzy-Systemen

Der Einsatz statischer Fuzzy-Systeme zur Fehlerisolation ist ein bekannter und vielfach verwendeter Ansatz [80]. Hierbei wird mithilfe eines SFS von den Residuenausschlägen auf den vorliegenden Fehlerfall geschlossen, wie Abbildung 7.5 (a) verdeutlicht. Die Auslegung des Fuzzy-Systems geschieht hierbei auf Basis einer Fehlersymptomtabelle, wie beispielhaft in Tabelle 7.1 zu sehen. Diese Tabelle zeigt die Symptome bei Auftreten eines bzw. mehrerer Fehler. Die Symptome können hierbei „positiv“, „null“ oder „negativ“ werden. Auf Basis dieser Tabelle kann dann durch Regeln der Form

$$\begin{aligned} &\text{Wenn } \rho_1 = L_{q_1}^{\rho_1} \text{ und } \dots \text{ und } \rho_m = L_{q_m}^{\rho_m}, \\ &\text{dann } \varphi_1 = L_{w_1}^{\varphi_1} \text{ und } \dots \text{ und } \varphi_l = L_{w_l}^{\varphi_l} \end{aligned}$$

auf den Fehler geschlossen werden. Hierbei ist $L_{q_v}^{\rho_v}$ der linguistische Wert der Abweichung des Symptoms ρ_v vom Nominalwert und $L_{w_r}^{\varphi_r}$ der Zuge-

Tabelle 7.1: Fehlersymptomtabelle mit drei Symptomen ρ_p für drei Fehlerfälle φ_i und die zugehörigen Doppelfehler φ_{ij} . Hierbei bezeichnet i den ersten, j den zweiten aufgetretenen Fehler sowie φ_0 den fehlerfreien Zustand. Die Symptome können positiv (+), negativ (-) oder null (0) werden.

	φ_0	φ_1	φ_2	φ_3	$\varphi_{12}, \varphi_{21}$	$\varphi_{13}, \varphi_{31}$	$\varphi_{23}, \varphi_{32}$
ρ_1	0	0	+	+	+	+	+
ρ_2	0	-	0	-	-	-	0
ρ_3	0	-	0	0	0	-	0

hörigkeitswert, dass der Fehlerfall φ_r eingetreten ist. Im Regelfall werden für die Symptome jeweils drei linguistische Werte $L_{1_v}^{\rho_v} = \text{„negativ“}$, $L_{2_v}^{\rho_v} = \text{„null“}$, $L_{3_v}^{\rho_v} = \text{„positiv“}$ verwendet. Die beiden linguistischen Werte $L_{1_r}^{\varphi_r} = \text{„kein Fehler“}$ und $L_{2_r}^{\varphi_r} = \text{„Fehler“}$ beschreiben die Fehlerzugehörigkeiten.

Die Möglichkeit, Fehler zu isolieren, hängt von den gewählten Symptomen ab. Dies soll am Beispiel der Fehlersymptomtabelle 7.1 verdeutlicht werden. Offensichtlich sind die Einzelfehler vollständig isolierbar, da eine eindeutige Zuordnung von Symptomen zu Fehlerfällen möglich ist. Betrachtet man allerdings zusätzlich auch die Doppelfehler, so ist keine eindeutige Zuordnung mehr möglich. Die Fehlerfälle φ_2 , φ_{32} und φ_{23} sowie φ_3 , φ_{21} und φ_{12} weisen jeweils dieselben Symptome auf und sind somit nicht unterscheidbar. Eine Isolation von Mehrfehlerfällen ist in diesem Fall also nicht sinnvoll möglich.

Über das Problem der Mehrfehlerisolation hinaus, gibt es eine weitere Gruppe von Fehlern, die nicht isolierbar sind. Dies sind Fehler, die einen dynamischen Symptom- bzw. Residuenverlauf hervorrufen, wie Abbildung 7.6 zeigt. Während das statische Residuum nach Eintritt des Fehlers einen bestimmten, konstanten Wert annimmt, verändert sich das dynamische Residuum über der Zeit. Typischerweise treten solche Residuenverläufe bei geregelten Systemen auf. Befindet sich eine Prozessgröße, die ein Residuum generiert, im Regelkreis, so wird sie nach Eintreten des Fehlers wieder eingeregelt. Dadurch geht auch das Residuum wieder auf Null zurück. Da die Symptomtabelle prinzipiell einen statischen Zusammenhang zwischen Fehlern und Residuen abbildet, können Fehler nicht korrekt erkannt werden. Vielmehr wird das System nach der Ausregelung wieder als fehlerfrei eingestuft.

Es zeigt sich also, dass das SFS bei dynamischen Residuen sowie im Mehrfehlerfall die Fehler nicht mehr eindeutig isolieren kann. Wie im folgenden Abschnitt gezeigt wird, kann allerdings mithilfe einer Erweiterung von SFS auf DRFS die Zahl der isolierbaren Fehler deutlich erhöht werden.

7.3.2 Fehlerisolation mit zeitdiskreten rekurrenten Fuzzy-Systemen

Der vorherige Abschnitt hat gezeigt, dass die Mehrfehlerisolation mit SFS selbst bei Symptomtabellen, die eine Isolation aller Einzelfehler erlauben, meist nicht eindeutig möglich ist. Ähnliches gilt für Fehler mit dynamischer Residuencharakteristik. Bezieht man dagegen Informationen über die dynamische Entwicklung der Symptome mit ein, so kann die Fehlerisolation

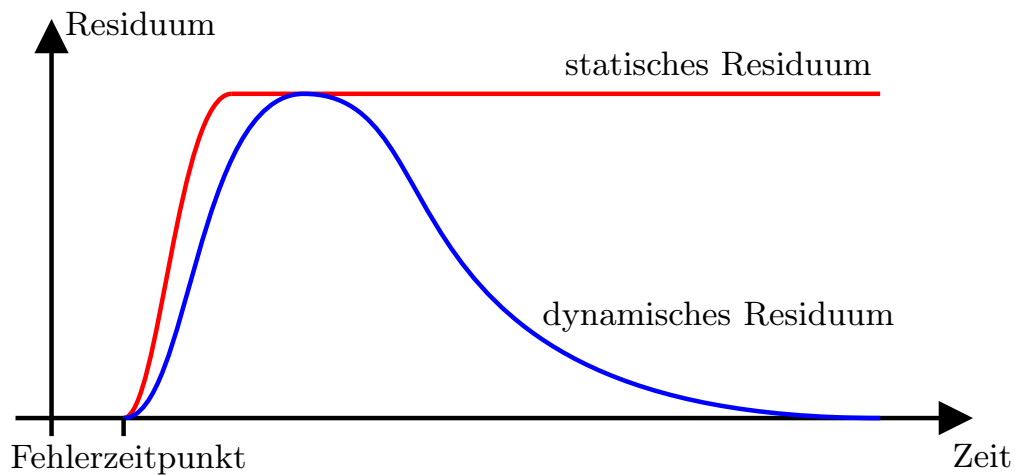


Abbildung 7.6: Beispiele für einen statischen und dynamischen Residuenverlauf.

deutlich verbessert werden. Dies wird durch die Erweiterung von SFS zu DRFS ermöglicht, die Abbildung 7.5 zeigt.

Hierbei werden die Fehlerfälle φ_r nicht mehr als Ausgänge eines SFS, sondern als Zustände des DRFS aufgefasst. Durch die Rückführung dieser Zustände, also der Fehlerzugehörigkeiten, werden dann die neuen Fehlerzugehörigkeiten auf Basis der Residuen und der alten Fehlerzugehörigkeiten bestimmt. Es ergeben sich nun Regeln der Form

$$\begin{aligned} &\text{Wenn } \varphi_1(k) = L_{j_1}^{\varphi_1} \quad \text{und } \dots \text{ und } \varphi_l(k) = L_{j_l}^{\varphi_l} \\ &\text{und } \rho_1(k) = L_{q_1}^{\rho_1} \quad \text{und } \dots \text{ und } \rho_m(k) = L_{q_m}^{\rho_m}, \\ &\text{dann } \varphi_1(k+1) = L_{w_1}^{\varphi_1} \text{ und } \dots \text{ und } \varphi_l(k+1) = L_{w_l}^{\varphi_l}. \end{aligned}$$

Das zugrunde liegende Fuzzy-System ist dabei ein DRFS, bei dem die Rückkopplungskorrespondenz nach Abschnitt 3.2 erfüllt ist. Abhängig von der Wahl der Zugehörigkeitsfunktionen für die Fehlerzustände, gibt es zwei Auslegungsformen des DRFS, die nun erläutert werden.

Aufbau des DRFS

Der erste Schritt zum Aufbau des DRFS besteht in der Festlegung der Zugehörigkeitsfunktionen. Hierbei sind zum einen die Zugehörigkeitsfunktionen der Eingangsgrößen, also der Symptome, zum anderen die Zugehörigkeitsfunktionen der Fehlerzustände zu definieren. Da die Symptome im Regelfall Abweichungen vom Nominalwert darstellen, ist es sinnvoll, die Zugehörigkeitsfunktionen symmetrisch zum Wert null aufzubauen. Üblicherweise werden die drei linguistischen Werte „negativ“, „null“ und „positiv“

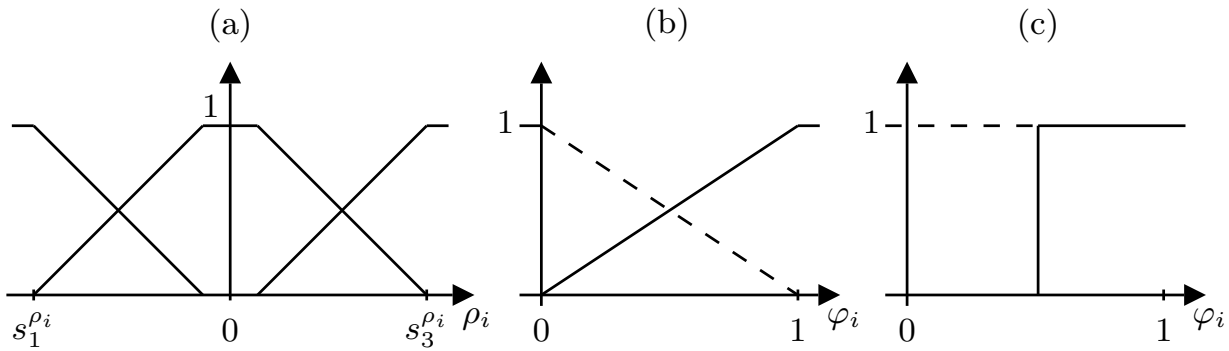


Abbildung 7.7: (a) Zugehörigkeitsfunktionen der Symptome, (b) Zugehörigkeitsfunktionen der Fehlerzustände bei DRFS, (c) Zugehörigkeitsfunktionen der Fehlerzustände bei hybridem DRFS.

tiv“ verwendet. Es sind aber auch mehr linguistische Werte denkbar, wenn man das Verhalten der Symptome genauer auflösen möchte. Da die Symptome im Regelfall rauschbehaftet sind, wird für den linguistischen Wert „null“ eine trapezförmige Zugehörigkeitsfunktion verwendet, sodass kleine Abweichungen vom Nominalwert keine Veränderung der Zugehörigkeit hervorrufen. Die Breite des Trapezes kann durch grobe Abschätzung der Varianz des Rauschens festgelegt werden. Abbildung 7.7 (a) illustriert dies. Für die Definition der Zugehörigkeitsfunktionen der Fehlerzustände bestehen grundsätzlich zwei Möglichkeiten. Die erste Möglichkeit ist die Wahl von zwei Rampenfunktionen, die der üblichen Definition DRFS mit Rückkopplungskorrespondenz entspricht (Abbildung 7.7 (b)). Diese Wahl führt dazu, dass die Fehlerzustände entsprechend Fehlerzugehörigkeiten zwischen null und eins annehmen. In einigen Anwendungen, z.B. bei der fehlertoleranten Regelung, möchte man dagegen eine eindeutige Aussage über die Existenz eines Fehlers haben. Verwendet man nun DRFS, so stellt sich die Frage, ab welchen Zugehörigkeitswerten ein Fehler als erkannt bzw. noch nicht erkannt eingestuft werden soll.

Um dieses Problem zu umgehen, kann man alternativ auch Sprungfunktionen als Zugehörigkeitsfunktionen verwenden, wie Abbildung 7.7 (c) zeigt. Dann können die Fehlerzustände nur noch Werte von null oder eins annehmen, also entweder den linguistischen Wert „fehlerfrei“ oder „Fehler“. Da die Symptome nach wie vor fuzzifiziert werden, wird diese Variante von DRFS in Anlehnung an [99] als hybrides DRFS bezeichnet. Der Vorteil dieser Variante ist die eindeutige Zuordnung von Fehlerzuständen, Nachteil ist eine geringere Robustheit gegenüber Modellungenauigkeiten und Störungen aufgrund der scharfen Grenzen in den Zuständen.

Eine Erweiterung des bisherigen Aufbaus ist die Berücksichtigung von prozessrelevanten Größen. Diese Größen, beispielsweise Eingangssignale, werden zusätzlich zu den Symptomen als Eingänge des RFS zur Fehlerisolation verwendet. Dies bietet den Vorteil, dass das RFS dann von den Eingangs- bzw. Prozessgrößen abhängige Regelbasen erhält. Beispielsweise können die Kernpositionen bei variierenden Eingangssignalen verändert werden, wodurch die Robustheit der Fehlerisolation gesteigert werden kann. Die Erweiterung um Prozessgrößen erlaubt so die Einführung von adaptiven Schwellen [80], die bereits bei der statischen Fehlerisolation erfolgreich eingesetzt werden. Nachteil dieser Erweiterung ist, dass sich die Regelbasis weiter vergrößert, wodurch sich der Entwurfsaufwand erhöht.

Auslegung der Regelbasis

Nachdem der Aufbau des DRFS festgelegt ist, soll nun die Auslegung der Regelbasis diskutiert werden. Ein prinzipieller Nachteil bei der Verwendung DRFS ist die im Vergleich zu SFS deutlich höhere Zahl an Regeln, die eingestellt werden müssen. Da die DRFS zur Fehlerisolation die Rückkopplungskorrespondenz aufweisen, besteht allerdings aufgrund der Nähe zu Zustandsautomaten die Möglichkeit, sich zur Auslegung der Fehlerisolation an einem solchen Zustandsautomaten zu orientieren. Da die Fehler als Zustandsgrößen des DRFS angesehen werden, kann man deren Kernpositionen als Zustände eines Automaten auffassen. Die Regelbasis definiert dann, ob und welcher Zustandsübergang bei einer bestimmten Symptomkonstellation eintritt. Einen Ausschnitt des sich aus Tabelle 7.1 ergebenden Graphen zeigt Abbildung 7.8. Mithilfe des Graphen, der sich aus dem qualitativen Wissen über die Fehlerauswirkungen aufstellen lässt, kann die Regelbasis des DRFS einfach hergeleitet werden. So ergibt sich beispielsweise für den Übergang von Fehler φ_3 zu Fehler φ_{32} die Regel:

Wenn $\varphi_1(k) = \text{„fehlerfrei“}$ und $\varphi_2(k) = \text{„fehlerfrei“}$ und $\varphi_3(k) = \text{„Fehler“}$
 und $\rho_1(k) = \text{„positiv“}$ und $\rho_2(k) = \text{„null“}$ und $\rho_3(k) = \text{„null“}$,
 dann $\varphi_1(k+1) = \text{„fehlerfrei“}$ und $\varphi_2(k+1) = \text{„Fehler“}$ und $\varphi_3(k+1) = \text{„Fehler“}$.

Allerdings ist die Auslegung nicht immer so eindeutig wie in der vorigen Regel. Insbesondere durch die Rückführung des Fehlerzustands ergeben sich weitere Regeln, deren Auslegung nicht eindeutig ist. Dies soll an einem einfachen Beispielsystem mit einem Residuum r mit den linguistischen Werten „niedrig“ und „hoch“ und einem Fehlerzustand φ mit den linguistischen

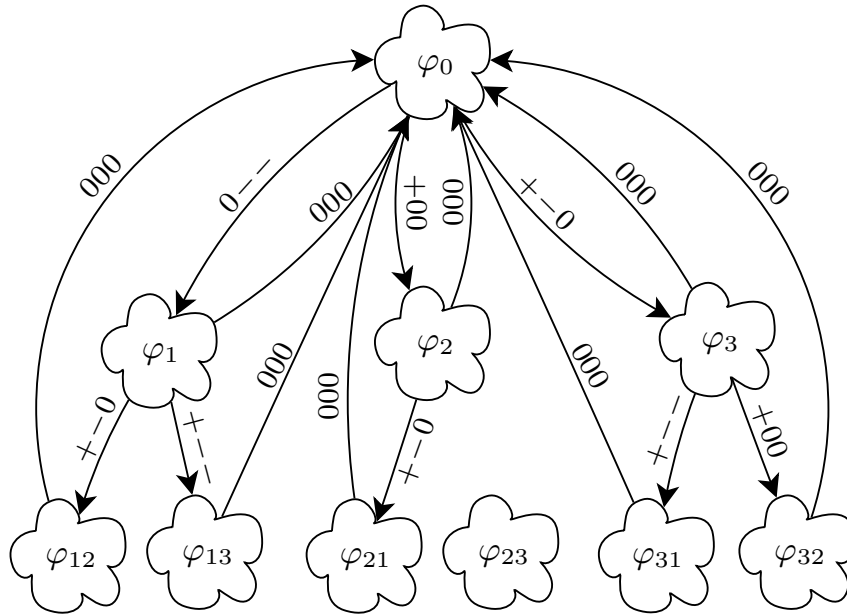


Abbildung 7.8: Darstellung der Fehlersymptomtabelle als linguistischer Zustandsautomat. Der Doppelfehler φ_{23} ist nicht isolierbar.

Werten „fehlerfrei“ und „Fehler“ genauer diskutiert werden. Es ergeben sich zunächst die vier Regeln:

Regel 1: Wenn $r(k) = \text{„null“}$ und $x(k) = \text{„fehlerfrei“}$, dann $x(k+1) = ?$,

Regel 2: Wenn $r(k) = \text{„hoch“}$ und $x(k) = \text{„fehlerfrei“}$, dann $x(k+1) = ?$,

Regel 3: Wenn $r(k) = \text{„null“}$ und $x(k) = \text{„Fehler“}$, dann $x(k+1) = ?$,

Regel 4: Wenn $r(k) = \text{„hoch“}$ und $x(k) = \text{„Fehler“}$, dann $x(k+1) = ?$,

bei denen die Konklusionen entsprechend ausgelegt werden sollen.

Regel 1 bildet den fehlerfreien Betrieb ab. Entsprechend ist der Fehlerzustand weiterhin fehlerfrei, es gilt also $x(k+1) = \text{„fehlerfrei“}$. Da der Fehlerzustand seinen linguistischen Wert beibehält, wird die Regel als *Halteregel* bezeichnet.

Ähnliches gilt für Regel 4. Hier befindet sich das DRFS bereits in dem zum Residuum „hoch“ passenden Fehlerzustand „Fehler“. Da der Fehler auch weiterhin angezeigt werden soll, gilt $x(k+1) = \text{„Fehler“}$. Auch dies ist eine Halteregel.

Regel 2 beschreibt den Fall, wenn der Fehler im System gerade aufgetreten ist. In diesem Fall nimmt das Residuum bereits den linguistischen Wert „hoch“ an, während der Fehlerzustand noch „fehlerfrei“ ist. Entsprechend sollte der Fehlerzustand im nächsten Zeitschritt den Fehler anzeigen, es soll

also $x(k+1)$ = „Fehler“ gelten. Da die Regel den Übergang vom fehlerfreien in den Fehlerzustand angibt, wird sie *Übergangsregel* genannt.

Im Gegensatz zu den vorigen Regeln ist die Schlussfolgerung der Regel 3 nicht eindeutig festgelegt. Vielmehr wird sie durch die Charakteristik der betrachteten Fehlerfälle und Voraussetzungen des Anwenders bestimmt. Prinzipiell wird mit Regel 3 der Fall berücksichtigt, bei dem bereits ein Fehler erkannt wurde, das Residuum aber wieder auf den linguistischen Wert „niedrig“ abgeklungen ist. Abhängig von der Anwendung sind nun verschiedene Schlussfolgerungen denkbar:

Die erste Möglichkeit besteht darin, als Folgezustand $x(k+1)$ = „fehlerfrei“ zu wählen. Dies ist dann sinnvoll, wenn ein Fehler nur sporadisch auftritt und wieder verschwinden kann. In diesem Fall wird Regel 3 in Anlehnung an [99] als *Rücksetzregel* verwendet. Auch bei stark rauschbehafteten Residuen, die zu einer irrtümlichen Erkennung von Fehlern führen, ist eine Rückkehr zum fehlerfreien Zustand wünschenswert. In diesem Fall ist die Regel aber keine Rücksetzregel im obigen Sinne. Vielmehr soll die Robustheit der Fehlererkennung erhöht werden, indem der linguistische Zustand „Fehler“ durch die Regel verringert bzw. abgeschwächt wird. Deshalb wird diese Auslegung im Folgenden als *Abschwächregel* bezeichnet.

Die zweite Möglichkeit ist, als Folgezustand $x(k+1)$ = „Fehler“ zu wählen. Hier sind zwei Fälle denkbar. Zum einen kann man bei einigen Fehlern davon ausgehen, dass sie ohne Reparatur nicht wieder verschwinden. Dann sollte der Fehlerzustand unabhängig vom Residuum gehalten werden. Zum anderen kann diese Auslegung bei dynamischen Fehlern notwendig sein, beispielsweise wenn das Residuum nach Eintritt des Fehlers ausgeregelt wird. Dann ist das Residuum „niedrig“ Teil des den Fehler beschreibenden Musters, sodass der Fehlerzustand gehalten werden sollte. Prinzipiell ist diese Auslegung eine Form der Halteregel. Weil sie aber außerdem den Zugehörigkeitswert des Zustands „Fehler“ erhöht und verstärkt, wird sie auch als *Verstärkerregel* bezeichnet.

Schließlich besteht auch die Möglichkeit, als Folgezustand $x(k+1)$ = „don't care“ zu wählen. Dies ist motiviert durch den Widerspruch zwischen dem scheinbar fehlerfreien Residuum und dem bereits erkannten Fehler. Durch die Wahl „don't care“ und damit eines neutralen Ausgangswerts, werden die Zugehörigkeitswerte des Fehlerzustands nicht beeinflusst. Diese Auslegung wird *neutrale Regel* genannt.

Zur Auslegung von Regel 3 ist also eine tiefere Kenntnis der Fehlercharakteristik und möglicher Störungen nötig. Werden mehrere Residuen und Fehler betrachtet, so treten entsprechend mehrere solcher Regeln auf,

bei denen die Konklusion abhängig von der Anwendung ausgelegt werden muss.

Abschließend sei noch darauf hingewiesen, dass der größte Teil der Regeln in der Regelbasis nicht festgelegt werden muss, da für diese Regeln keine Fehlerfälle definiert sind. Die Ausgangszugehörigkeitsfunktion dieser Regeln wird entsprechend auf den Wert Null gesetzt, sodass diese Regeln nicht zur Berechnung des neuen Fehlerzustands beitragen.

Automatisierte Auslegung des Fehlerisolators

Trotz der im vorigen Abschnitt beschriebenen Möglichkeit, sich bei der Auslegung der Regelbasis an Automaten zu orientieren, kann die Zuordnung von linguistischen Fehlerwerten sehr zeitaufwändig sein. Dies gilt insbesondere dann, wenn viele Fehler gleichzeitig isoliert werden sollen, da die Zahl der Regeln exponentiell mit der Zahl der Fehler ansteigt. Da die hier vorgestellte Systemstruktur zur Fehlerisolation der von DRFS mit Rückkopplungskorrespondenz entspricht, kann man die in Abschnitt 6.2 beschriebene Methodik zur Identifikation der Regelbasis auch für die Auslegung des Fehlerisolators verwenden, sodass die Auslegung automatisiert erfolgen kann.

Betrachtet man das Optimierungsproblem 6.1 bzw. 6.2 zur Identifikation der Regelbasis, so ist lediglich der allgemeine Zustand \mathbf{x} durch den Fehlerzustand φ sowie der Systemeingang \mathbf{u} durch die Residuen \mathbf{r} zu ersetzen. Wir gehen also davon aus, dass Datensätze der Form

$$(\hat{r}_p^{(v)}(k), \hat{\varphi}_i^{(v)}(k), \hat{\varphi}_i^{(v)}(k+1)),$$

die die betrachteten Fehlerfälle und ihre Residuencharakteristik enthalten, zur Auslegung der Regelbasis vorliegen. Dies setzt voraus, dass entsprechende Fehler in das System eingebracht werden und der resultierende Residuenverlauf aufgezeichnet wird. Eine solche Generierung der Datensätze wird in der Praxis für bestimmte Fehlerfälle nicht möglich sein, da der Fehler unerwünschte Betriebszustände auslösen kann. In solchen Fällen können die Datensätze dann mithilfe von Expertenwissen oder Ergebnissen eines Simulationsmodells um diese Fehlerfälle ergänzt werden. Bei der Erstellung der Datensätze ist des Weiteren zu beachten, dass die Fehlerfälle nur einen geringen Teil der Regeln in der Regelbasis tatsächlich definieren. Der größte Teil der Regeln ist durch die Fehlerfälle nicht abgedeckt. Die Regelbasis ist also durch die Datensätze nicht vollständig definiert. Entsprechend sollte Optimierungsproblem 6.2, das eine nicht vollständig definierte Regelbasis berücksichtigt, verwendet werden. Eine andere Möglichkeit, die

Unvollständigkeit der Regelbasis zu berücksichtigen, besteht darin, auch nicht definierte Regeln in den Datensätzen zu definieren. Dies ist einfach realisierbar, indem allen Regeln, die nicht durch die Fehlerfälle definiert sind, nachträglich der Wert „don't care“, bzw. null zugewiesen wird. In der Praxis hat sich gezeigt, dass sich die Ergebnisse der Optimierung deutlich verbessern, wenn mehr Regeln durch die Datensätze definiert werden.

Vorteile bei dynamischen Fehlern und Mehrfehlern

Die Erweiterung von SFS zu DRFS war motiviert durch die Tatsache, dass SFS nicht in der Lage sind, dynamische Fehler sowie mehrere nacheinander auftretende Fehler zu isolieren. Diese Schwierigkeit verschwindet bei der Verwendung von DRFS, wie im Weiteren gezeigt wird.

Zunächst werden Fehler mit dynamischer Residuencharakteristik betrachtet (vgl. Abbildung 7.6). Werden die Residuen nun fuzzifiziert, so ist nicht mehr allein der linguistische Wert nach Eintritt des Fehlers relevant, sondern die Reihenfolge, in der die linguistischen Werte aufeinander folgen. So wird das Residuum nach Auftreten des Fehlers zunächst mit der Zugehörigkeit eins zum linguistischen Wert "positiv" fuzzifiziert. Klingt das Residuum anschließend langsam ab, so verringert sich die Zugehörigkeit wieder bis das Residuum schließlich mit Zugehörigkeit eins zum linguistischen Wert "null" fuzzifiziert wird. Es entsteht also das Muster "null", "positiv", "null" von aufeinander folgenden linguistischen Werten. Solche Muster können, wie bereits in [99] bei der sequentiellen Mustererkennung beschrieben, durch das DRFS nachgebildet werden. Allgemein können in dieser Weise beliebige Residuendynamiken in der Fehlerdiagnose berücksichtigt werden, indem der Residuenverlauf in ein Muster von linguistischen Werten umgewandelt wird, dem dann vom DRFS ein Fehlerfall zugeordnet wird.

Auch bei der Isolation mehrerer Fehler ergibt sich eine deutliche Verbesserung der Fehlerisolation. Hierzu betrachten wir in der Fehlersymptomtabelle 7.1 beispielhaft den Mehrfehlerfall φ_{12} , der mithilfe des SFS nicht vom Einzelfehler φ_3 unterschieden werden kann. Ist bereits der Fehler φ_1 aufgetreten und korrekt isoliert worden, so wird diese Information zurückgeführt. Bei einer Änderung der Symptome auf "+ - 0" und der zusätzlichen Information, dass der Fehler φ_1 bereits erkannt wurde, kann dann auf den Doppelfehler φ_{12} geschlossen werden. Der Einzelfehler φ_3 ist so ausgeschlossen. Ähnliches geschieht, wenn zuerst Fehler φ_2 auftritt. Ein Sonderfall sind die Doppelfehler φ_{23} bzw. φ_{32} . Offensichtlich hängt die Isolierbarkeit von der Reihenfolge ab, in der die Fehler auftreten. Tritt zuerst φ_3 mit den Symptomen "+ - 0" auf, ist die Isolation des Doppelfehlers φ_{32} durch die

Änderung der Symptome auf "+00" möglich. Tritt dagegen zuerst φ_2 mit den Symptomen "+00" auf, so kann der Doppelfehler φ_{23} mit den gleichen Symptomen "+00" nicht erkannt werden, da sich die Symptome nicht verändert haben. Insgesamt kann die Zahl der isolierbaren Fehler durch die Erweiterung auf DRFS deutlich erhöht werden.

Vergleich mit Automaten

Wie die letzten Betrachtungen schon nahelegen, verhalten sich das DRFS und ein entsprechend ausgelegter Zustandsautomat ähnlich. Würden binäre Zugehörigkeitsfunktionen sowohl für die Zustände als auch die Eingänge verwendet, wären beide Systeme äquivalent. Das DRFS hat allerdings dann Vorteile, wenn die Symptome aufgrund von nicht modellierten Dynamiken oder Rauschen von ihren nominellen Werten abweichen. Um dies zu illustrieren, wird wieder die Symptomtabelle 7.1 betrachtet. Wir gehen davon aus, dass das Symptom ρ_2 bei Auftreten von Fehler φ_3 aufgrund von Modellierungsungenauigkeiten nicht so stark negativ wird wie erwartet. Die harte Grenze von "null" nach "negativ" wird dadurch beim Zustandsautomaten nicht überschritten, das DRFS befindet sich im Interpolationsbereich zwischen den Kernpositionen "negativ" und "null". Der Zustandsautomat wird aufgrund der Symptome "+00" auf den falschen Fehlerfall φ_2 schließen. Das DRFS dagegen wird aufgrund der unscharfen Grenzen auch Fehler φ_3 mit einer bestimmten Fehlerzugehörigkeit anzeigen. Tritt nun zusätzlich Fehler φ_1 auf, so kann der Zustandsautomat diesen nicht anzeigen, da nach der Erkennung von φ_2 der Doppelfehler φ_{31} nicht erreicht werden kann (vgl. Abbildung 7.8). Da das DRFS den Fehler φ_3 mit einer bestimmten Zugehörigkeit zurückführt, isoliert das DRFS den Doppelfehler φ_{31} dagegen korrekt.

Insgesamt ist das DRFS somit deutlich weniger anfällig gegenüber Störungen und Modellungenauigkeiten als der Zustandsautomat. Dies führt auch dazu, dass die bei Automaten oftmals schwierige Festlegung der Schwellwerte, also der Residuenwerte, bei denen ein Zustandsübergang stattfindet, aufgrund der unscharfen Übergänge des DRFS abgemildert wird, was die Auslegung vereinfacht. Hierauf wird im folgenden Abschnitt anhand eines Anwendungsbeispiels nochmals genauer eingegangen.

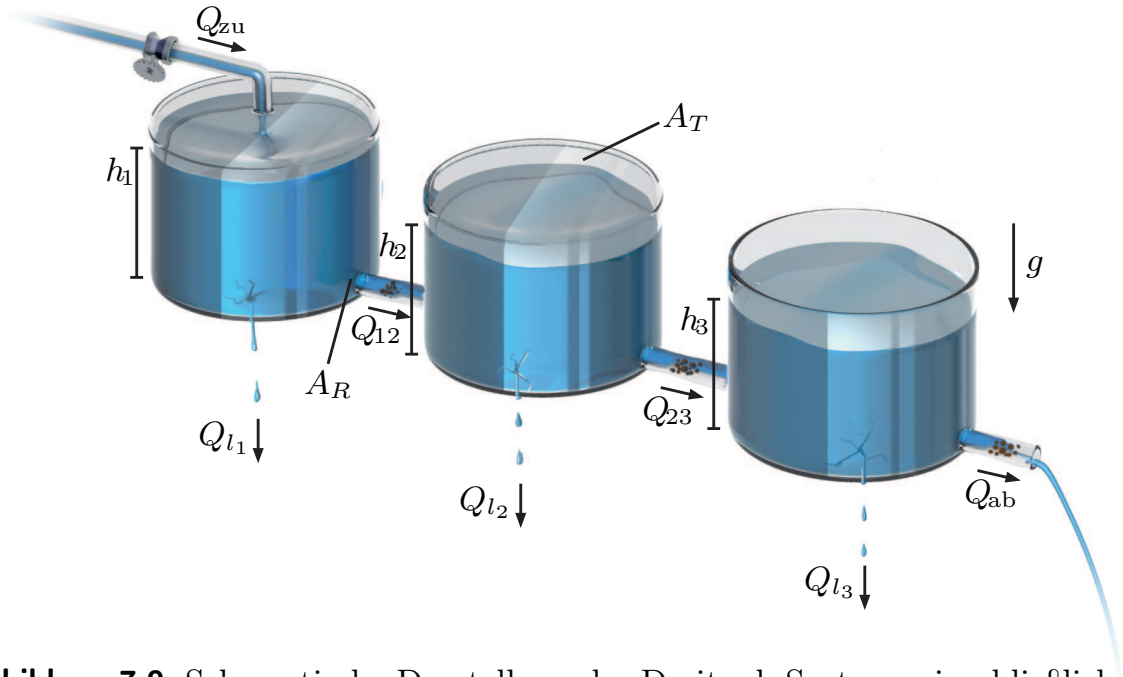


Abbildung 7.9: Schematische Darstellung des Dreitank-Systems einschließlich der Ventile zur Veranschaulichung der Fehlereinflüsse. Der Volumenstrom Q_{l_i} fließt im Falle eines Lecks der Größe A_{l_i} aus dem Tank i ab. Eine Verstopfung des Ausmaßes $v_{\mu\nu}$ beeinflusst die zwischen den Tanks fließenden Volumenströme $Q_{\mu\nu}$.

7.4 Beispiel: Dreitank-System

Der vorgestellte Ansatz zur FDI soll im Folgenden anhand des Dreitank-Benchmarksystems [151] erläutert werden, das in Abbildung 7.9 illustriert ist. Die Modellbildung des Systems erfolgt über die Bilanzgleichungen für die drei Tanks.

Verwendet man die Füllstände h_i als Zustandsgrößen x_i , ergibt sich das Zustandsraummodell

$$\begin{aligned} A_T \dot{x}_1 &= Q_{zu} - Q_{12}, \\ A_T \dot{x}_2 &= Q_{12} - Q_{23}, \\ A_T \dot{x}_3 &= Q_{23} - Q_{ab}, \end{aligned} \tag{7.2}$$

mit den Volumenströmen

$$Q_{\mu\nu} = c_{\mu\nu} A_{R_{\mu\nu}} \cdot \text{sign}(x_\mu - x_\nu) \sqrt{2g \cdot |x_\mu - x_\nu|}$$

zwischen Tank μ und Tank ν bzw. dem Volumenstrom im Abfluss

$$Q_{ab} = c_{ab} A_{R_{ab}} \sqrt{2g \cdot x_3}.$$

Hierbei sind $c_{\mu\nu} = 1$ und $c_{ab} = 0,8$ Durchflussparameter sowie $A_T = 0,0157\text{m}^2$ bzw. $A_{R_{\mu\nu}} = 5 \cdot 10^{-5}\text{m}^2$ Tank- bzw. Rohrquerschnitte.

Zur Validierung der vorgestellten Methoden zur FDI werden die in Abbildung 7.9 gezeigten Fehlerfälle untersucht. Hierbei handelt es sich zum einen um Leckagen ($\varphi_1, \dots, \varphi_3$) in den drei Tanks, die als zusätzliche Abflüsse

$$Q_{l_i} = c_{ab} A_{l_i}(t) \sqrt{2g \cdot x_i}$$

mit den zeitvarianten Leckquerschnitten $A_{l_i} \in [0, A_R)$ in den Bilanzgleichungen (7.2) modelliert werden. Zum anderen werden Verstopfungen in den Verbindungsrohren zwischen Tank 1 und Tank 2 (φ_4) bzw. Tank 2 und Tank 3 (φ_5) sowie im Abfluss (φ_6) durch variierende Querschnittsflächen

$$A_{R_{\mu\nu}}(t) = A_R(1 - v_{\mu\nu}(t)) \quad \text{mit } v_{\mu\nu}(t) \in [0, 1)$$

berücksichtigt.

Der erste Schritt zur FDI besteht in der Modellierung des Dreitank-Systems durch ein DRFS. Zur Generierung von Trainings- und Validierungsdaten wird das Modell mit einer Abtastzeit von $T_0 = 5\text{s}$ diskretisiert und mit amplitudenmodulierten Pseudo-Rausch-Binär-Signalen angeregt (Abbildung B.6 im Anhang). Mithilfe der so gewonnenen Datensätze und den Verfahren aus Kapitel 5 wird nun ein DRFS aufgebaut, welches das Dreitank-System sehr genau nachbilden kann. Die Residuen werden dann auf Basis dieses Modells und der Struktur aus Abbildung 7.4 bestimmt. Die Fehlersymptome ρ_i werden schließlich gemäß

$$\rho_i = \frac{r_i - r_{0,i}}{u}, \quad u > 0$$

mit den Residuen $r_{0,i}$ im fehlerfreien Zustand gebildet. Letztere werden durch Vergleich von Dreitank-System und RFS-Modell gemäß Gl. (7.1) gebildet. Durch die Normierung auf die Eingangsgröße u reagieren die Symptome weniger empfindlich auf Schwankungen der Eingangsgröße. Als Ergebnis ergibt sich die Fehlersymptomtabelle 7.2, die alle Einzelfehler und Doppelfehler beinhaltet. Die mit Fragezeichen gekennzeichneten Einträge der Tabelle zeigen an, dass das Residuum bei Doppelfehlern abhängig von den jeweiligen Fehlergrößen sowohl positiv als auch negativ werden kann.

Betrachtet man nur die Einzelfehler, so zeigt sich, dass die Residuen aufgrund der FDI-Struktur nur sensitiv auf Fehler sind, die direkt auf den jeweiligen Zustand wirken. Entsprechend können alle Einzelfehler durch ein SFS isoliert werden. Weiterhin können aufgrund der Symptomverteilung auch einige Doppelfehler durch ein SFS isoliert werden. Problematisch ist

Tabelle 7.2: Fehlersymptomtabelle für das Dreitank-System. Als Fehlerfälle werden die Leckagen in den Tanks ($\varphi_1, \dots, \varphi_3$) und die Verstopfungen ($\varphi_4, \dots, \varphi_6$) sowie alle Doppelfehler φ_{ij} betrachtet. Die Doppelfehler φ_{ji} haben die gleichen Symptome wie φ_{ij} und sind deshalb nicht explizit aufgeführt. Den fehlerfreien Zustand bezeichnet φ_0 . Zur Herleitung der Symptomtabelle wurde $v_{\mu\nu} = 0,25$ und $A_{l_i} = 0,3A_R$ verwendet.

	φ_0	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_{12}	φ_{13}	φ_{14}	φ_{15}
ρ_1	0	-	0	0	+	0	0	-	-	?	-
ρ_2	0	0	-	0	-	+	0	-	0	-	+
ρ_3	0	0	0	-	0	-	+	0	-	0	-

	φ_{16}	φ_{23}	φ_{24}	φ_{25}	φ_{26}	φ_{34}	φ_{35}	φ_{36}	φ_{45}	φ_{46}	φ_{56}
ρ_1	-	0	+	0	0	+	0	0	+	+	0
ρ_2	0	-	-	?	-	-	+	0	?	-	+
ρ_3	+	-	0	-	+	-	-	?	-	+	?

dagegen beispielsweise der Doppelfehlerfall φ_{24} . Dieser hat dieselbe Symptomcharakteristik wie der Einzelfehler φ_4 , sodass das SFS diesen Fall nicht isolieren kann. Verwendet man dagegen ein DRFS, so ist der Doppelfehler eindeutig isolierbar, wenn der Fehler φ_2 zuvor erkannt wurde. Tabelle 7.3 stellt die Isolierbarkeit der Einzel- und Doppelfehler für das SFS und DRFS zusammen. Hierbei lassen sich verschiedene Klassen unterscheiden:

- Weiß hinterlegt: Diese Fehler sind sowohl mit dem SFS als auch mit dem DRFS isolierbar.
- Grün hinterlegt: Diese Fehler sind nicht mit dem SFS, aber mit dem DRFS isolierbar.
- Orange hinterlegt: Diese Fehler sind mit dem SFS nicht isolierbar. Die Isolierbarkeit mit dem DRFS hängt von den Fehlergrößen ab. Ist die Auswirkung des zweiten Fehlers hinreichend groß, ist der Doppelfehler isolierbar. Ist die Auswirkung des zweiten auftretenden Fehlers auf das Residuum dagegen deutlich kleiner als die des ersten, so ist der Doppelfehler nicht isolierbar. Der erste Fehler verdeckt den zweiten.

Tabelle 7.3: Übersicht über die Isolierbarkeit der Fehlerfälle. 22 der 36 Fehlerfälle können durch ein SFS isoliert werden (nicht hinterlegt). Durch das DRFS können vier weitere vollständig (grün) und sechs teilweise (orange) isoliert werden. Rot hinterlegte Fehlerfälle können nicht isoliert werden.

	2. Fehler					
	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6
1. Fehler						
φ_1	φ_1	φ_{12}	φ_{13}	φ_{14}	φ_{15}	φ_{16}
φ_2	φ_{21}	φ_2	φ_{23}	φ_{24}	φ_{25}	φ_{26}
φ_3	φ_{31}	φ_{32}	φ_3	φ_{34}	φ_{35}	φ_{36}
φ_4	φ_{41}	φ_{42}	φ_{43}	φ_4	φ_{45}	φ_{46}
φ_5	φ_{51}	φ_{52}	φ_{53}	φ_{54}	φ_5	φ_{56}
φ_6	φ_{61}	φ_{62}	φ_{63}	φ_{64}	φ_{65}	φ_6

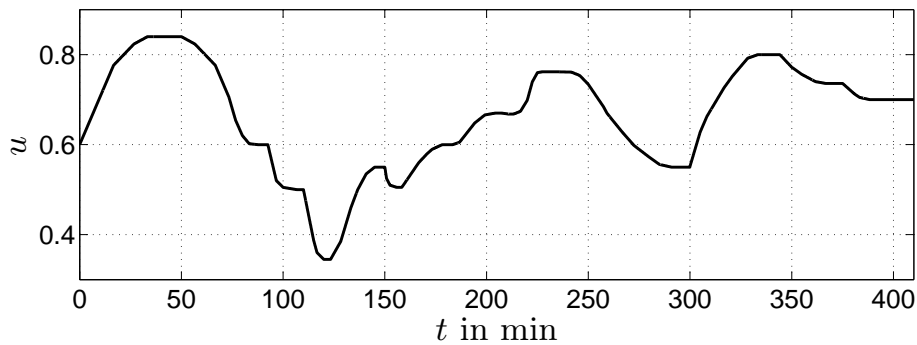


Abbildung 7.10: Anregungssignal für den Dreitank-Prozess (Testsignal).

- Rot hinterlegt: Diese Fehler sind weder mit dem SFS noch mit dem DRFS isolierbar.

Insgesamt lassen sich sechs Einzelfehler sowie weitere 16 Doppelfehler mithilfe der Residuenstruktur nach Abbildung 7.4 bereits mit einem SFS isolieren. Durch die Verwendung eines DRFS lassen sich vier weitere Fälle vollständig sowie sechs Fälle abhängig von der Fehlergröße isolieren.

Abschließend sollen zwei Beispielszenarien zur FDI gezeigt werden. Im ersten Szenario soll das DRFS mit einem SFS verglichen werden. Hierzu wird das Drei-Tank-System mit dem Eingangssignal aus Abbildung 7.10 beaufschlagt. Im zunächst fehlerfreien Betrieb tritt zum Zeitpunkt $t = 50$ min Fehler φ_2 , d.h. eine Leckage in Tank 2 auf. Nach Beseitigung dieses Fehlers tritt bei $t = 150$ min wieder eine Leckage in Tank 2 auf. Zusätzlich verstopft bei $t = 170$ min das Rohr zwischen Tank 1 und 2 (Fehler φ_4).

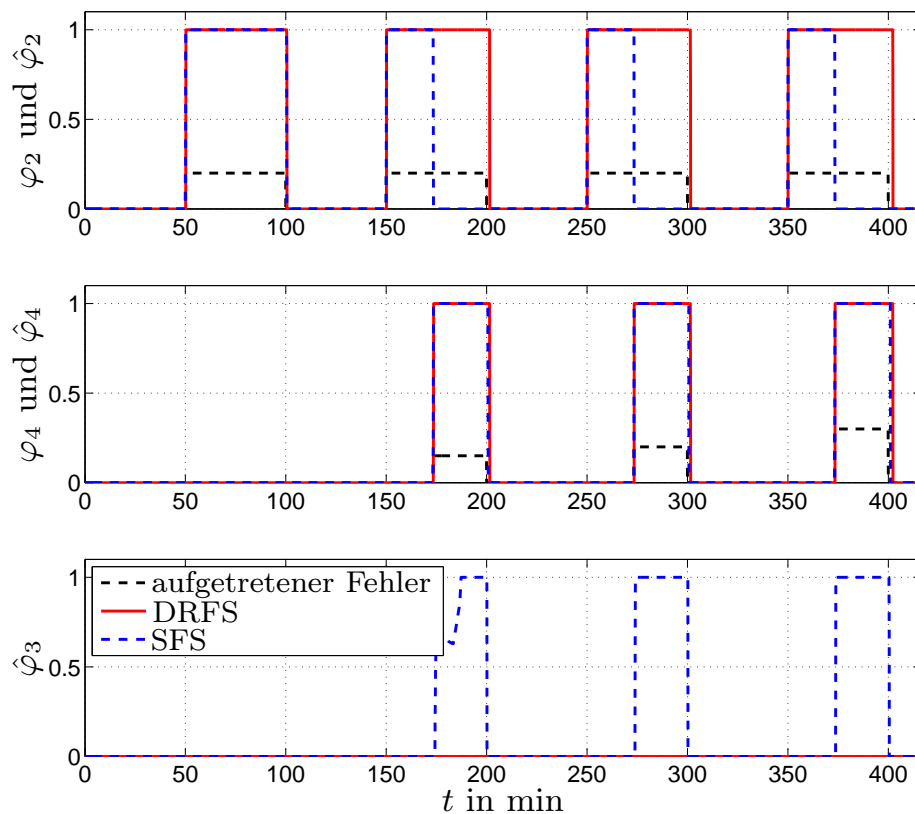


Abbildung 7.11: Ergebnisse der Fehlerisolation mit SFS und DRFS (Testsituation). Während das SFS den Fehler $\hat{\varphi}_3$ fälschlicherweise erkennt, detektiert das DRFS alle Fehlerfälle korrekt.

Beide Fehler werden bei $t = 200$ min behoben. Dieser Vorgang wird zweimal mit zunehmenden Größen der Verstopfung wiederholt. Abbildung 7.11 zeigt die Ergebnisse der FDI mit SFS und DRFS. Sowohl das SFS als auch das DRFS können den Einzelfehler φ_2 korrekt isolieren. Wie anhand der Fehlersymptomtabelle zu erwarten, kann das SFS den Doppelfehlerfall φ_{24} allerdings nicht isolieren. Vielmehr wird fälschlicherweise der Doppelfehler φ_{34} angezeigt. Das DRFS kann dagegen unabhängig von den Fehlergrößen die jeweiligen Fehlerfälle richtig isolieren.

Im zweiten Szenario wird das DRFS mit einem Zustandsautomaten verglichen. Letzterer wurde so ausgelegt, dass die harten Zustandsübergänge auf dem Mittelwert der zwei zugehörigen Kernpositionen des DRFS liegen. Im Szenario tritt zum Zeitpunkt $t = 20$ min Fehler φ_5 mit einer Verstopfung von 5% zwischen Tank 2 und 3 auf. Aufgrund der sehr kleinen Fehlergröße wird das Symptom ρ_2 weniger stark positiv als angenommen. Zum Zeitpunkt $t = 80$ min verstopft dann zusätzlich der Abfluss (Fehler φ_6). Abbildung 7.12 zeigt die Ergebnisse für den Zustandsautomaten und das

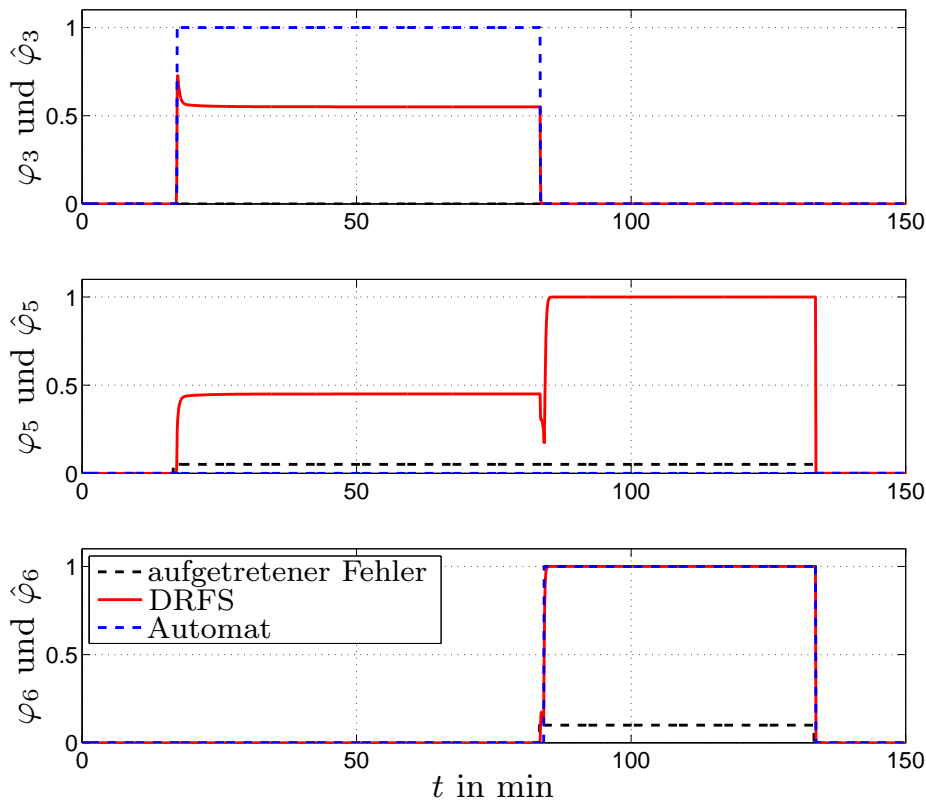


Abbildung 7.12: Ergebnisse der Fehlerisolation mit Zustandsautomat und DRFS. Der Automat erkennt fälschlicherweise φ_3 , während das DRFS die Fehler φ_3 und φ_5 mit den Fehlerzugehörigkeiten 0,55 bzw. 0,45 erkennt. Beim zusätzlichen Auftreten von Fehler φ_6 erkennt der Automat zunächst keinen Fehler und später nur Fehler φ_6 . Der Fehler φ_5 wird dagegen gar nicht erkannt. Das DRFS kann dagegen den Doppelfehler φ_{56} korrekt isolieren.

DRFS. Aufgrund der auftretenden Symptome erkennt der Zustandsautomat fälschlicherweise Fehler φ_3 , während das DRFS beide möglichen Fehler φ_3 und φ_5 mit den Fehlerzugehörigkeiten 0,55 bzw. 0,45 erkennt. Tritt nun der zweite Fehler auf, wird der Doppelfehler φ_{56} vom DRFS mit Zugehörigkeit 1 richtig isoliert. Der Zustandsautomat wechselt dagegen zunächst in den fehlerfreien Zustand und isoliert dann lediglich den Fehler φ_6 . Der Fehler φ_5 wird gar nicht erkannt. Dies veranschaulicht die größere Robustheit DRFS im Vergleich zu Zustandsautomaten bei Modellungenauigkeiten wie der beschriebenen geringeren Fehlergröße.

7.5 Zusammenfassung und Fazit

In diesem Kapitel wurden Verfahren zur Fehlerdiagnose mit RFS vorgestellt. Hierbei wurde sowohl die modellbasierte Residuengenerierung mit RFS als auch die Fehlerisolation behandelt. Letztere basiert auf einer Erweiterung statischer Fuzzy-Systeme. Hierbei weisen rekurrente Fuzzy-Systeme insbesondere bei dynamischen sowie mehreren nacheinander auftretenden Fehlern deutliche Vorteile gegenüber statischen Fuzzy-Systemen auf. Des Weiteren erlauben RFS im Vergleich zu Automaten eine deutlich robustere Fehlerisolation bei Modellunsicherheiten oder nicht genau bekannten Fehlercharakteristiken.

Prinzipbedingt entstehen bei der Verwendung von DRFS deutlich größere Regelbasen als bei SFS, da die Zustände zusätzlich zurückgeführt werden. Dies führt auf einen höheren Entwurfsaufwand, da insbesondere durch die Rückführung zusätzliche Regeln entstehen, die nicht direkt ersichtlich den Ablauf der Fehlerisolation beeinflussen. Durch die Verwendung der Modellierungsverfahren für DRFS ist allerdings auch eine automatisierte Auslegung möglich, durch die sich der Entwurfsaufwand wiederum reduziert.

8 Anwendungen

Nachdem in den vorigen Kapiteln neue *Methoden* zur Modellierung und Fehlerdiagnose mithilfe rekurrenter Fuzzy-Systeme vorgestellt wurden, beschäftigt sich dieses Kapitel nun mit der konkreten *Anwendung* auf praktische Probleme. Solche Anwendungsbeispiele sind insbesondere deshalb von großer Bedeutung, da den bisher betrachteten Simulationsbeispielen einige vereinfachende Annahmen zugrunde liegen, die in der Praxis oft nicht erfüllt sind. Dies betrifft beispielsweise die Vernachlässigung von Mess- und Prozessrauschen, das in allen realen Anwendungen auftritt. Des Weiteren lässt sich das Anregungssignal bei realen Prozessen oft nicht beliebig vorgeben. Vielmehr werden die Daten während des Betriebs aufgenommen, sodass nur ein bestimmter Betriebsbereich abgedeckt wird und Einflüsse von unterlagerten Regelkreisen das Systemverhalten beeinflussen. Entsprechend ist es notwendig, die Effektivität der Verfahren auch in realen Systemen zu untersuchen.

Rekurrente Fuzzy-Systeme wurden bereits bei verschiedenen Anwendungen erfolgreich eingesetzt. Die erste Anwendung zeitdiskreter rekurrenter Fuzzy-Systeme beschäftigte sich mit der Erkennung von Durchbrüchen in Stranggießanlagen [2, 97]. Dieses System, bei dem das RFS als sequentieller Mustererkenner eingesetzt wird, wird weltweit in der industriellen Praxis eingesetzt. Des Weiteren wird in [97] eine auf zeitdiskreten RFS basierende Verkehrssimulation vorgestellt. Auch Modellierungsprobleme sind bereits mit RFS gelöst worden. So wird das elektrische Verhalten einer Brennstoffzelle mittels zeitkontinuierlicher RFS modelliert [48, 49]. Allerdings werden hier keine automatisierten Verfahren, sondern ausschließlich Expertenwissen mit anschließender händischer Optimierung verwendet. Die Ergebnisse zeigen bereits die Eignung KRFS zur Nachbildung dynamischer Prozesse.

In den folgenden Abschnitten soll die Methodik zur Modellierung dynamischer Systeme und zur Fehlerdiagnose an realen Systemen getestet werden. So werden in Abschnitt 8.1 RFS zur Nachbildung des dynamischen Verhaltens einer Destillationskolonne eingesetzt. In Abschnitt 8.2 wird die Fehlerdiagnose mit RFS am Beispiel einer hydraulischen Servoachse mit Duplex-Ventilsystem untersucht.

8.1 Modellierung einer Destillationskolonne

Destillationskolonnen werden in der chemischen Industrie zur Trennung zweier oder mehrerer Stoffe eingesetzt. Ziel ist hierbei, Stoffe mit möglichst hohem Reinheitsgrad zu erhalten. Um eine hohe Produktqualität zu gewährleisten, werden deshalb neben herkömmlichen PID-Reglern zunehmend modellbasierte Regelungsverfahren eingesetzt. Letztere benötigen allerdings ein möglichst genaues Modell des dynamischen Verhaltens der Destillationskolonne. Deshalb wurden in den letzten Jahren verschiedene Modelle entwickelt, die sich sowohl im Detaillierungsgrad als auch in der Form der Modellbildung unterscheiden.

Sehr genaue, gleichzeitig aber auch sehr komplexe Modelle von Destillationskolonnen wurden in [42, 43] mithilfe physikalischer Modellbildung erzeugt, die insbesondere die Modellierung der Massen- und Energiebilanzen für jede Trennstufe der Kolonne umfassen. Es ergibt sich schließlich ein Deskriptorsystem mit über hundert Zustandsgrößen. Solch umfassende Modelle sind allerdings für viele Anwendungen zu komplex, sodass in [180] verschiedene Möglichkeiten der Vereinfachung angegeben werden, die schließlich auf ein lineares Modell führen. Dieses wurde als Grundlage verschiedener Regelungsverfahren verwendet [180]. Allerdings werden in obigen Modellen die Stoffkonzentrationen als Regelgrößen definiert. Bei der Verwendung der Modelle wird somit die teure und mit großen Messverzögerungen verbundene Konzentrationsmessung vorausgesetzt. Um dies zu vermeiden, werden in den meisten Destillationskolonnen anstatt der Konzentrationen die Temperaturen in den Kolonnenböden als Ersatzgrößen verwendet, da sie ein ähnliches zeitliches Verhalten wie die Konzentrationen aufweisen. Entsprechend werden an Stelle der Konzentrationen entweder direkt die Temperaturen geregelt [75, 102, 116] oder die Konzentrationen mithilfe der Temperaturen geschätzt [133].

Einen alternativen Weg zur Modellierung des Verhaltens von Destillationskolonnen stellen datengestützte Methoden dar, wobei bisher zu meist statische und dynamische neuronale Netze [71] und statische Fuzzy-Systeme [118, 128] eingesetzt werden. Diese können zum einen die physikalische Modellbildung ergänzen, beispielsweise zur Modellierung von Nichtlinearitäten oder zur Modellierung von Zusammenhängen, die physikalisch nicht beschrieben werden können [14, 118]. Zum anderen kann aber auch das Gesamtverhalten durch ein neuronales Netz dargestellt werden [129, 179]. Der Vorteil besteht dann darin, dass man sich die oft langwierige Herleitung der physikalischen Gleichungen sowie deren Parametrierung spart. Stattdessen wird allein auf Basis von Messdatensätzen ein

Modell generiert. Nachteil ist allerdings, dass das Verhalten nicht mehr transparent nachgebildet wird.

Einen ebenfalls datengestützten Ansatz verfolgt die im Folgenden vorgestellte Modellierung einer Destillationskolonne mithilfe zeitdiskreter rekurrenter Fuzzy-Systeme. Allerdings steht im Gegensatz obigen Approximationsmodellen die qualitative Modellierung des Systemverhaltens im Vordergrund, sodass das Modell linguistisch interpretierbar und damit deutlich transparenter als neuronale Netze ist. Die zur Modellierung eingesetzten Algorithmen sind in Kapitel 5 und 6 eingehend beschrieben. Abschnitt 8.1.1 stellt zunächst die Funktionsweise der Destillationskolonne sowie die gewonnenen Datensätze vor. Im anschließenden Abschnitt 8.1.2 werden dann die Ergebnisse der Modellierung mit KRFS präsentiert.

8.1.1 Systembeschreibung

Die Destillation ist ein thermischer Trennprozess, bei dem ein Stoffgemisch aufgrund unterschiedlicher Siede- und Taupunkte der Komponenten separiert wird [46, 70]. Zur industriellen Destillation werden Destillationskolonnen eingesetzt. Abbildung 8.1 zeigt den Aufbau der im Folgenden betrachteten Destillationskolonne. Sie lässt sich in die drei wesentlichen Bereiche Kolonnensumpf, Turm und Kolonnenkopf aufteilen. Im Kolonnensumpf ist ein Verdampfer angebracht, über den ein Teil des schwerer siedenden Stoffes wieder in den Kolonnenturm zurückgeführt wird. Eine ähnliche Aufgabe hat der Kondensator im Kolonnenkopf, mit dessen Hilfe der leichter siedende Stoff verflüssigt und im Behälter aufgefangen wird. Über das Rücklaufventil kann ein Teil des Leichtsieders wieder in den Kolonnenturm zurückgeführt werden. Außerdem wird auf etwa halber Höhe über ein Ventil das Stoffgemisch kontinuierlich zugeführt.

Die eigentliche Destillation findet im Turm statt. Der Kolonnenturm enthält eine Vielzahl von Kolonnenböden, die mit Öffnungen versehen sind. Über diese kann Dampf von unten eindringen, während gleichzeitig die flüssigen Bestandteile des Gemisches über den Bodenrand nach unten abfließen. Es entsteht somit ein Gegenstrom von Dampf und Flüssigkeit, wobei jeweils ein Wärme- und Stoffaustausch stattfindet. Der schwersiedende Stoff kondensiert aus dem Dampf in die Flüssigkeit. Durch die dabei entstehende Kondensationswärme verdampft der leichtsiedende Stoff. Entsprechend reichert sich der Dampf mit dem leichtsiedenden Stoff an, wobei die Temperatur mit der Strömungsrichtung abnimmt. Umgekehrt reichert sich der schwersiedende Stoff in der Flüssigkeit an, wobei die Temperatur zunimmt. Durch die Rückführungen des sich unten absetzenden Schwersie-

zum einen der Dampfdruck P_1 und zum anderen die Ventilstellung V_2 gemessen. Da der Dampfdruck durch eine Regelung nahezu konstant gehalten wird, kann die im Verdampfer zugeführte Wärmemenge weitgehend über die Ventilstellung bestimmt werden. Im Kolonnenkopf werden der Abfluss F_3 und Rücklauf F_2 des Leichtsieders sowie das Rücklaufverhältnis F_4 gemessen. Des Weiteren werden die Füllstände in der Kolonne L_1 und im Behälter L_2 gemessen, die allerdings während der Messdatenaufnahme auf einen konstanten Wert geregelt werden. In der Kolonne selbst werden die Temperaturen $T_1 - T_5$ an fünf verschiedenen Punkten sowie der Differenzdruck P_D über der Kolonne gemessen.

Zur Modellierung des dynamischen Verhaltens wurden mithilfe der beschriebenen Sensorik Messdatensätze¹ über einen längeren Zeitraum aufgenommen. Bei der Verwendung dieser Daten ist allerdings zu beachten, dass sie während des Betriebs der Anlage, also am geregelten System aufgenommen wurden. Entsprechend spiegeln die Messdaten nur die abgefahrenen Betriebsbereiche wieder. Ebenso sind die Anregungssignale durch den Betrieb bestimmt und können nicht beliebig vorgegeben werden. Außerdem werden während der Datenaufnahme zur Erhaltung der Prozessqualität die Füllstände in Behälter und Kolonne konstant geregelt. Des Weiteren wird die Temperatur T_4 geregelt, wodurch auch die Messdatenverläufe beeinflusst werden.

8.1.2 Modellierung der Kolonne mit RFS

Nachdem die Funktionsweise der Destillationskolonne beschrieben ist, soll nun das dynamische Verhalten mithilfe eines zeitdiskreten RFS mit Rückkopplungskorrespondenz nachgebildet werden. Ziel ist hierbei, ein Modell zu entwickeln, das neben einer guten Nachbildung des dynamischen Verhaltens insbesondere auch qualitative Informationen über den Prozess liefert. Dies umfasst zum einen die Relevanz der betrachteten Eingangs- und Zustandsgrößen und zum anderen die linguistische Beschreibung der Systemdynamik in Form einer Regelbasis. Da der Fokus der Modellierung vor allem auf der qualitativen Modellierung liegt, werden die zur Verfügung stehen-

¹Die Datensätze wurden an einer Destillationskolonne der Firma Evonik-Degussa aufgenommen. Für die Bereitstellung dieser Daten möchte ich mich bei Evonik, insbesondere bei Dr. Anna Flemming und Dr. Kai Dadhe herzlich bedanken. Aufgrund von Auflagen zur Geheimhaltung dürfen einige Informationen zum Prozess, beispielsweise Abtastzeiten, nicht öffentlich gemacht werden. Des Weiteren sind die Datensätze teilweise normiert.

den Daten entsprechend der Ausführungen in Kapitel 6 mit $T_A^* = 500 \cdot T_A$ abgetastet, wobei T_A die ursprüngliche Abtastzeit ist.

Zunächst sollen aber mithilfe von Vorwissen aus den zur Verfügung stehenden Messgrößen die Eingangs- und Zustandsgrößen ausgewählt werden. Als erste Eingangsgröße kann prinzipiell der Rücklauf F_2 aus dem Auffangbehälter in den Kolonnenturm benutzt werden. Allerdings zeigte sich im Verlauf der Optimierung, dass die Änderung des Rückflusses $\Delta F_2 = F_2(k) - F_2(k-1)$ mehr Information beinhaltet, sodass letztere als Eingangsgröße verwendet wird. Außerdem werden der Zufluss in die Destillationskolonne, der über die Ventilstellung V_1 eingestellt wird, sowie die Änderung der Ventilstellung $\Delta V_1 = V_1(k) - V_1(k-1)$ als weitere Eingangsgrößen festgelegt.

Als Zustandsgrößen können prinzipiell die an fünf verschiedenen Stellen im Turm gemessenen Temperaturen sowie der über der Kolonne anliegende Differenzdruck verwendet werden. Allerdings ergeben sich einige praktische Einschränkungen. Da die Temperatur T_4 während des Betriebs der Anlage auf einen konstanten Wert geregelt wird, also an nur einem Arbeitspunkt betrieben wird, ist sie zur Modellierung ungeeignet. Dies führt auch dazu, dass die Anregung durch die im Erhitzer eingebrachte Wärmemenge relativ gering ist. Zusätzlich ist die Temperatur T_5 über weite Bereiche des Datensatzes nahezu konstant, was die Modellierung zusätzlich erschwert. Deshalb wird auf die Modellierung des Verhaltens im unteren Teil der Kolonne verzichtet. Im Kolonnenkopf stellt sich die Temperatur T_1 etwa auf die Siedetemperatur des leichter siedenden Stoffes ein und ist somit nahezu konstant. Deshalb wird alternativ nicht die absolute Temperatur im Kolonnenkopf, sondern die Temperaturdifferenz zwischen Kopftemperatur T_1 und der etwas tiefer liegenden Temperatur T_2 verwendet. Dies ist auch deshalb eine sinnvolle Wahl, da, wie u.a. in [116] gezeigt wird, die Steilheit des Temperaturprofils ein Indikator für die sich einstellenden Konzentrationen ist. Entsprechend werden also die Differenztemperatur $\Delta T_{12} = T_1 - T_2$, die Temperaturen T_2 und T_3 sowie der Differenzdruck P_D als Zustandsgrößen definiert. Damit ergibt sich ein Zustandsraummodell 4. Ordnung mit dem Zustandsvektor $\mathbf{x} = [\Delta T_{12}, T_2, T_3, P_D]^T$ und dem Eingangsvektor $\mathbf{u} = [\Delta F_2, V_1, \Delta V_1]^T$.

Auswahl der relevanten Zustands- und Eingangsgrößen

Mit diesen Festlegungen kann nun die Nachbildung mit DRFS erfolgen. Hierzu wird in mehreren Schritten vorgegangen. Aufgrund der vergleichsweise hohen Systemdimension ergibt sich, wie bereits beschrieben, bereits

Tabelle 8.1: Ergebnisse der Variablenauswahl für das dynamische Modell der Destillationskolonne. Das Ergebnis gibt jeweils die Zahl der Hyperquader an, in denen die Variable als relevant eingestuft wird. Die ausgewählten Größen sind jeweils grün unterlegt.

	$\Delta T_{12}(k)$	$T_2(k)$	$T_3(k)$	$P_D(k)$	$\Delta F_2(k)$	$V_1(k)$	$\Delta V_1(k)$
$\Delta T_{12}(k+1)$	7/7	4/7	5/7	3/7	6/7	2/7	0/7
$T_2(k+1)$	4/7	6/7	6/7	1/7	6/7	2/7	0/7
$T_3(k+1)$	3/7	6/7	7/7	2/7	1/7	7/7	5/7
$P_D(k+1)$	6/7	1/7	3/7	6/7	2/7	4/7	5/7

bei wenigen Zugehörigkeitsfunktionen pro Dimension eine sehr hohe Zahl an Regeln. Um die erhaltene Regelbasis so kompakt und einfach wie möglich zu halten, ist es sinnvoll, die Zahl der Regeln zu verringern. Hierzu wird im ersten Schritt mithilfe von Auswahlalgorithmus 5.5 die Relevanz der Eingangs- und Zustandsgrößen auf die einzelnen Zustände untersucht. Im zweiten Schritt wird dann die Regelbasis des DRFS mithilfe der ganzzahligen quadratischen Optimierung identifiziert. Um gute Startwerte für die Zahl an Zugehörigkeitsfunktionen und die Lage der Kernpositionen zu erhalten, wird mithilfe von Algorithmus 5.2 eine initiale Partitionierung des Eingangsraums vorgegeben. Außerdem wird die Regelbasis auf redundante Regeln untersucht. Im letzten Schritt werden die freien Parameter optimiert.

Zunächst wird nun die Variablenauswahl betrachtet. Hierzu wurde der Auswahlalgorithmus 5.6 in sieben Hyperquadern, die die in den Datensätzen abgebildeten Arbeitsbereiche der Kolonne möglichst gut nachbilden, angewendet. Zur Ermittlung dieser Hyperquader werden zunächst einige Iterationen von Algorithmus 5.2 durchgeführt, sodass die Struktur des DRFS, d.h. die Partitionierung des Eingangsraums, zur Modellierung ausreichend genau ist. Anschließend werden die Hyperquader, die die größte Zahl an Datenpaaren enthalten, ausgewählt.

Die Ergebnisse der Variablenauswahl sind in Tabelle 8.1 angegeben. Da der Rücklauf aus dem Kondensator F_2 am Kolonnenkopf einfließt, beeinflusst ΔF_2 hauptsächlich die Differenztemperatur ΔT_{12} und die Temperatur T_2 . Ebenso wirkt der Zufluss sowie die Änderung des Zuflusses auf die Temperatur T_3 . Der direkte Einfluss der jeweiligen Eingangsgröße auf die anderen Temperaturen ist dagegen gering. Außerdem beeinflussen sich die

benachbarten Temperaturen gegenseitig. Da der qualitative Einfluss von ΔT_{12} und T_2 aufgrund der nahezu konstanten Temperatur T_1 sehr ähnlich ist, wird die Temperatur T_2 zur Modellierung der Differenztemperatur nicht verwendet. Sie beinhaltet im Vergleich zu ΔT_{12} keine zusätzliche Information. Ähnliches gilt für die Modellierung von T_2 . Ebenso wird zur Modellierung von T_3 nur die Ventilstellung V_1 , nicht aber die Änderung der Ventilstellung ΔV_1 berücksichtigt. Für den Differenzdruck wird vor allem die Temperaturdifferenz im Kopf als relevant eingestuft. Dies lässt sich dadurch erklären, dass der Differenzdruck mit dem Temperaturprofil korreliert ist. Letzteres wird durch die Temperaturdifferenzen abgebildet, wodurch sich die hohe Relevanz von ΔT_{12} erklärt. Außerdem wirkt der Zufluss bzw. dessen Änderung auf P_D . Beide weisen ähnliche qualitative Informationen auf, allerdings erwies sich ΔV_1 als besser geeignet, den Differenzdruck nachzubilden und wird deshalb als Eingang verwendet.

Die Tatsache, dass die Relevanz nicht in allen Hyperquaden festgestellt werden kann, liegt in der Beschaffenheit der Datensätze und der teilweise nur geringen Anregung begründet. So wird ΔF_2 in einem Hyperquader als nicht relevant für T_2 eingestuft, weil sich in dem betrachteten Betriebsbereich ΔF_2 kaum ändert. Ähnliches tritt auch bei anderen Größen auf. Durch die Betrachtung verschiedener Hyperquader kann dies aber ausgeglichen werden. Mit den obigen Ergebnissen wird nun das allgemeine Zustandsraummodell des dynamischen Verhaltens der Destillationskolonne gemäß

$$\mathbf{x}(k+1) = \begin{pmatrix} f_1(\Delta T_{12}, T_3, \Delta F_2) \\ f_2(T_2, T_3, \Delta F_2) \\ f_3(T_2, T_3, V_1) \\ f_4(\Delta T_{12}, P_D, \Delta V_1) \end{pmatrix} = \begin{pmatrix} f_1(x_1, x_3, u_1) \\ f_2(x_2, x_3, u_1) \\ f_3(x_2, x_3, u_2) \\ f_4(x_1, x_4, u_3) \end{pmatrix} = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))$$

angenommen. Diese Systemstruktur stellt die Basis zur Modellierung des dynamischen Verhaltens mit DRFS dar. Entsprechend werden zur qualitativen Modellbildung, d.h. zur Ermittlung der Regelbasis, nur die als relevant eingestuften Variablen verwendet.

Identifikation der Regelbasis

Im Folgenden wird das DRFS sukzessive aufgebaut, indem die Dynamik der Zustände einzeln modelliert wird. Dies ist möglich, da die Regelbasis für jeden Zustand unabhängig von der Festlegung der Regelbasis der anderen Zustände identifiziert werden kann. Die Parameteroptimierung erfolgt dann allerdings unter Berücksichtigung aller Zustände.

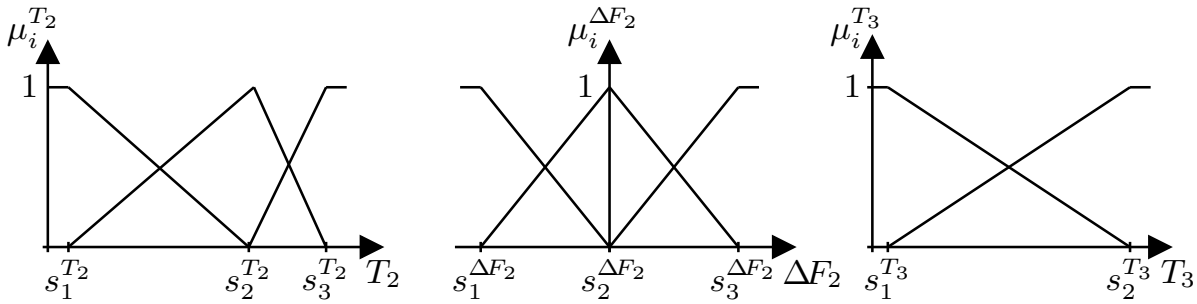


Abbildung 8.2: Zugehörigkeitsfunktionen der relevanten Größen T_2 , T_3 und ΔF_2 zur Modellierung von T_2 . Ähnliche Zugehörigkeitsfunktionen werden auch zur Modellierung der anderen Zustandsgrößen verwendet.

Bevor allerdings die Regelbasis identifiziert werden kann, muss die Zahl der Zugehörigkeitsfunktionen und die Lage der zugehörigen Kernpositionen bekannt sein. Hierbei kann man auf zweierlei Art vorgehen. Zum einen kann man die Zahl und Lage heuristisch mithilfe von Vorwissen festlegen. Da zur qualitativen Modellierung im Regelfall nur eine geringe Zahl an Zugehörigkeitsfunktionen verwendet wird, ist eine solche heuristische Festlegung oft eine ausreichend gute Näherung. Die Kernpositionen werden dann meist äquidistant platziert. Allerdings zeigt sich, dass sich bei einigen Systemen die Qualität und Plausibilität der identifizierten Regelbasis durch die Eingangspartitionierung stark ändern kann. Um eine bessere Näherung zu erhalten, wird deshalb im Folgenden Algorithmus 5.2 zur Partitionierung des Eingangsraums für zeitdiskrete RFS angewendet. Dieser liefert zum einen einen sehr guten Anhaltspunkt zur Festlegung der Zahl an Zugehörigkeitsfunktionen. Zum anderen werden automatisch initiale Kernpositionen erzeugt, die die Nichtlinearität des Prozesses besser berücksichtigen.

Zur Modellierung der Temperatur T_2 wird zunächst die Zahl der Zugehörigkeitsfunktionen mit Algorithmus 5.2 festgelegt. Es ergeben sich drei Zugehörigkeitsfunktionen für T_2 und ΔF_2 sowie zwei Zugehörigkeitsfunktionen für T_3 , die in Abbildung 8.2 gezeigt werden. Die initialen äußeren Kernpositionen werden auf die minimalen und maximalen Werte im Datensatz festgelegt. Die dritte Kernposition des Zustands T_2 liegt bei 65% des Maximalwerts, die Kernpositionen sind also nicht äquidistant platziert. Werden die Kernpositionen äquidistant festgelegt, ergibt sich bei der anschließenden Identifikation der Regelbasis ein deutlich schlechteres und teilweise unplausibles Ergebnis.

Die Ergebnisse der Identifikation der Regelbasis mithilfe von Algorithmus 6.1 sind in Abbildung 8.3 zu sehen. Aus dieser linguistischen Darstel-

$\Delta F_2(k)$				$\Delta F_2(k)$					
$T_2(k+1)$		-	0	+	$T_2(k+1)$		-	0	+
$T_2(k)$	n	h	n	m	$T_2(k)$	n	dc	n	n
	m	m	m	m		m	h	m	n
	h	m	h	n		h	h	h	dc
$T_3(k) = „n“$					$T_3(k) = „h“$				

Abbildung 8.3: Regelbasis für die Zustandsgröße T_2 . Die Abkürzungen bzw. Symbole bedeuten n=„niedrig“, m=„mittel“, h=„hoch“, -=„negativ“, 0=„null“, +=„positiv“ und dc=„don't care“.

lung der Dynamik lassen sich einige Systemeigenschaften ablesen. Grundsätzlich wird die qualitative Dynamik der Temperatur T_2 maßgeblich von der Änderung des Rückflusses ΔF_2 bestimmt. Hat beispielsweise die Änderung im Rückfluss den linguistischen Wert „null“, so bleibt der Zustand im bisherigen linguistischen Wert (mittlere Spalte der Regelbasis). Ist die Änderung „positiv“, so wird die Temperatur tendenziell geringer. Dies ist auch sinnvoll, da die Temperatur des zurückgeführten kondensierten Leichtsieders geringer ist als die Temperatur am Rücklauf. Umgekehrt wird die Temperatur höher, wenn die Änderung „negativ“ ist. Der Einfluss der Temperatur T_3 , die unterhalb von Temperatur T_2 gemessen wird, ergibt sich aus einem Vergleich des linken und rechten Teils der Regelbasis. Offenbar wirken sich Änderungen im Rückfluss bei niedriger Temperatur T_3 weniger stark aus als bei hoher Temperatur T_3 . Da der zufließende Stoff kälter ist als die Temperatur in der Kolonne, wirkt sich ein veränderter Zufluss bei höherer Temperatur in der Kolonne etwas stärker aus.

Mithilfe von Algorithmus 6.1 wird die Regelbasis vollständig definiert, d.h., jeder Regel wird eine Ausgangszugehörigkeitsfunktion zugeordnet. Allerdings können einige Regeln, wie bereits in Abschnitt 5.3.3 beschrieben, redundant sein, können also auch durch andere Regeln beschrieben werden. Des Weiteren ergeben sich aufgrund der vorgegebenen gitterähnlichen Partitionierung Regeln, die durch den Datensatz kaum abgedeckt sind. Dann macht die Zuordnung eines linguistischen Ausgangswerts keinen Sinn. Zur Ermittlung solcher Regeln wird nun Algorithmus 5.4 auf die Regelbasis für T_2 angewendet. Hierbei werden zwei Regeln erkannt, bei denen sich der Approximationsfehler verringert, wenn die Ausgangszugehörigkeitsfunktion durch eine Interpolation der benachbarten Kernpositionen ersetzt wird.

$\Delta F_2(k)$				$\Delta F_2(k)$			
$\Delta T_{12}(k+1)$	-	0	+	$\Delta T_{12}(k+1)$	-	0	+
n	n	n	h	n	n	n	h
$\Delta T_{12}(k)$ m	m	m	m	$\Delta T_{12}(k)$ m	n	m	h
h	n	h	dc	h	dc	h	h
$T_3(k) = \text{„n“}$				$T_3(k) = \text{„h“}$			

Abbildung 8.4: Regelbasis für die Zustandsgröße ΔT_{12} . Die Abkürzungen bzw. Symbole bedeuten n=„niedrig“, m=„mittel“, h=„hoch“, -=„negativ“, 0=„null“, +=„positiv“ und dc=„don’t care“.

Diese Regeln werden in Abbildung 8.3 durch „dc = don’t care“ gekennzeichnet. Es handelt sich hier um Regeln, bei denen die Temperatur T_2 „niedrig“ und der Eingang ΔF_2 „negativ“ bzw. die Temperatur „hoch“ und der Eingang „positiv“ ist. Beide Fälle sind aber durch die Datensätze nicht abgedeckt und physikalisch auch nicht sinnvoll. Somit kann die Regelbasis also um zwei Regeln reduziert werden, sodass sich insgesamt 16 Regeln zur Beschreibung von T_2 ergeben.

Die Modellierung der Temperaturdifferenz ΔT_{12} ist der Modellierung der Temperatur T_2 sehr ähnlich, da sich aufgrund der nahezu konstanten Temperatur T_1 das qualitative Verhalten nur im Vorzeichen unterscheidet. Entsprechend ändert sich im Wesentlichen nur die Zuordnung der linguistischen Werte „niedrig“ und „hoch“. Die resultierende Regelbasis zeigt Abbildung 8.4.

Für die Modellierung der Temperatur T_3 werden zunächst drei linguistische Werte für T_3 und V_1 sowie zwei linguistische Werte für T_2 festgelegt. Die Kernpositionen für T_3 werden äquidistant gesetzt. Dagegen liegt $s_2^{V_1}$ bei 40% des Maximalwerts. Die optimierte Regelbasis zeigt Abbildung 8.5. Aus der Regelbasis ist deutlich zu erkennen, dass der Zufluss die Temperatur maßgeblich beeinflusst. Ist der Zufluss „hoch“, so steigt auch die Temperatur T_3 an. Ist er dagegen „niedrig“, sinkt die Temperatur ab. Für einen „mittleren“ Zufluss bleibt der Zustand erhalten. Der Einfluss der Temperatur T_2 ist vergleichsweise gering, wie ein Vergleich der Regelbasen für $T_2 = \text{„n“}$ und $T_2 = \text{„h“}$ zeigt. Allerdings liegen bei einer höheren Temperatur T_2 die linguistischen Werte der Temperatur T_3 über die gesamte Regelbasis etwas höher. Wie schon zuvor, werden wiederum redundante Regeln gefunden, die durch Interpolationsregeln ersetzt werden. Auffällig ist die relativ

$V_1(k)$				$V_1(k)$			
$T_3(k+1)$	n	m	h	$T_3(k+1)$	n	m	h
n	dc	n	n	n	n	m	dc
$T_3(k)$ m	n	m	h	$T_3(k)$ m	m	m	h
h	m	dc	dc	h	n	h	h
$T_2(k) = „n“$				$T_2(k) = „h“$			

Abbildung 8.5: Regelbasis für die Zustandsgröße T_3 . Die Abkürzungen bedeuten n=„niedrig“, m=„mittel“, h=„hoch“ und dc=„don't care“.

große Zahl an redundanten Regeln. Dies ist durch die Beschaffenheit des Datensatzes bedingt, die nur einen vergleichsweise kleinen Teil des durch die Kernpositionen gebildeten Arbeitsbereichs abdecken. Insbesondere der Bereich für niedrige Temperaturen ist nur wenig abgedeckt. Die schlechte Abdeckung ist bei der Modellierung von T_3 besonders ausgeprägt. Bei den anderen Zustandsgrößen ist die Abdeckung deutlich besser.

Des Weiteren wird der Differenzdruck modelliert. Hierzu werden drei linguistische Werte für den Druck P_D und die Differenztemperatur ΔT_{12} sowie zwei linguistische Werte für die Zuflussänderung ΔV_1 festgelegt. In Abbildung 8.6 ist die optimierte Regelbasis dargestellt. Wie schon zuvor entspricht die qualitative Nachbildung dem zu beobachtenden Systemverhalten. Prinzipiell führt eine Abnahme des Zuflusses zu einer Absenkung des Differenzdrucks, während eine Zunahme des Zuflusses zu einer Erhöhung von P_D führt. Entsprechend sind die linguistischen Werte im aktuellen Zeitschritt für $\Delta V_1(k) = „-“$ tendenziell niedriger als die linguistischen Werte im vorigen Zeitschritt. Umgekehrt verhält es sich für $\Delta V_1(k) = „+“$. Abweichungen von diesem grundsätzlichen Verhalten zeigen die beiden Regeln

Wenn $P_D(k) = „m“, \Delta T_{12}(k) = „h“$ und $\Delta V_1(k) = „-“$, dann $P_D(k+1) = „h“$,
 Wenn $P_D(k) = „h“, \Delta T_{12}(k) = „m“$ und $\Delta V_1(k) = „+“$, dann $P_D(k+1) = „n“$.

In diesen Regeln macht sich der Einfluss der Differenztemperatur stark bemerkbar. Tritt eine hohe Differenztemperatur ΔT_{12} auf, so führt dies unabhängig von der Änderung des Zuflusses zu einem ebenfalls hohen Differenzdruck. Bei Absinken der Differenztemperatur auf den linguistischen Wert „mittel“, sinkt dann auch der Differenzdruck wieder ab. Zusammenfassend wird ein niedriger und mittlerer Differenzdruck hauptsächlich von

$\Delta T_{12}(k)$				$\Delta T_{12}(k)$			
$P_D(k+1)$	n	m	h	$P_D(k+1)$	n	m	h
n	n	n	n	n	m	m	h
$P_D(k)$ m	m	n	h	$P_D(k)$ m	m	h	h
h	h	m	m	h	h	n	h
$\Delta V_1(k) = \text{„-“}$				$\Delta V_1(k) = \text{„+“}$			

Abbildung 8.6: Regelbasis für die Zustandsgröße P_D . Die Abkürzungen bedeuten n=„niedrig“, m=„mittel“, h=„hoch“, -=„negativ“ und +=„positiv“.

$\Delta V_1(k)$ beeinflusst. Ein hoher Differenzdruck tritt dagegen im Zusammenhang mit einer hohen Temperaturdifferenz auf.

Zusammenfassend ergibt sich also ein qualitatives Modell in Form eines DRFS mit vier Zustands- und drei Eingangsgrößen, das plausibel im Sinne der zu beobachtenden Systemdynamik ist. Die Systemdynamik wird hierbei durch insgesamt 64 Regeln zuzüglich acht „don’t care“-Regeln beschrieben. Es ergibt sich also angesichts der Systemdimension eine sehr kompakte Regelbasis, die leicht interpretierbar ist und somit auch einfach durch Anwender plausibilisiert werden kann.

Optimierung der Parameter

Nachdem die qualitative Modellbildung abgeschlossen ist, wird abschließend eine Optimierung der Parameter des DRFS, d.h. der Kernpositionen durchgeführt werden. Hierzu wird der Datensatz in Trainings- und Validierungsdaten aufgeteilt. Dies geschieht folgendermaßen. Da die Daten während des Betriebs aufgenommen wurden, bei dem verschiedene Arbeitsbereiche abgefahren werden, ist es nicht möglich, den vorhandenen Datensatz zu einem Zeitpunkt zu teilen und den einen Teil als Trainings- und den anderen Teil als Validierungsdatensatz zu verwenden. Die Trainingsdaten decken dann einen anderen Bereich ab als die Validierungsdaten. Deshalb wurde der Datensatz derart an verschiedenen Zeitpunkten geteilt, dass vergleichbare Arbeitsbereiche abgefahren werden. Mithilfe der Datensätze wird gemäß den Ausführungen in Abschnitt 6.3 ein Gradientenverfahren eingesetzt. Dieses wird abgebrochen, sobald der Validierungsfehler ansteigt. Die sich ergebenden Kernpositionen sind in Tabelle B.5 im Anhang angegeben. Die Verläufe der Eingangssignale Rückfluss und Zufluss

Tabelle 8.2: Ergebnisse der Modellierung der Destillationskolonne. Angegeben ist jeweils der RMSE und der maximale Fehler.

	Trainingsfehler		Validierungsfehler	
	RMSE	MAX	RMSE	MAX
x_1	0,029	0,359	0,041	0,407
x_2	0,054	0,578	0,052	0,596
x_3	0,046	0,493	0,053	0,607
x_4	0,013	0,067	0,018	0,072

werden in Abbildung 8.7 gezeigt. Die gemessenen Verläufe der Zustandsgrößen sowie die mittels des DRFS modellierten Verläufe für den gesamten Datensatz sind in Abbildung 8.8 zu sehen. Trotz des qualitativen Charakters des DRFS mit Rückkopplungskorrespondenz können die Verläufe insgesamt auch quantitativ gut nachgebildet werden. An einigen Stellen ist zu erkennen, dass das DRFS aufgrund der Abtastung schnelle Dynamiken nicht abbilden kann. Außerdem weist das DRFS bei Temperatur T_3 in einigen Abschnitten ein Offset gegenüber den gemessenen Werten auf. Dies ist auf die schon angesprochene schlechte Abdeckung des Trainingsdatensatzes zurückzuführen. Außerdem sind in Tabelle 8.2 der RMSE und der maximale Fehler für den Trainings- und Validierungsdatensatz gegeben. Wie ein Vergleich zeigt, sind die Fehler von Trainings- und Validierungsdatensatz ähnlich, sodass keine Überanpassung an die Trainingsdaten zu beobachten ist.

8.1.3 Fazit

Dieser Abschnitt zeigte die Modellierung des dynamischen Verhaltens einer Destillationskolonne mittels zeitdiskreter rekurrenter Fuzzy-Systeme. Hierbei wurden zum einen die zur Modellierung relevanten Zustands- und Eingangsgrößen identifiziert und zum anderen eine qualitative Modellbildung durchgeführt. Letztere führt auf ein vergleichsweise einfaches Modell, das in Form einer Regelbasis bzw. eines linguistischen Automaten dargestellt werden kann. Aufgrund der geringen Zahl an Zugehörigkeitsfunktionen ist die Regelbasis sehr kompakt und weist nur wenige leicht verständliche Regeln auf. Die Ergebnisse der abschließenden quantitativen Modellierung

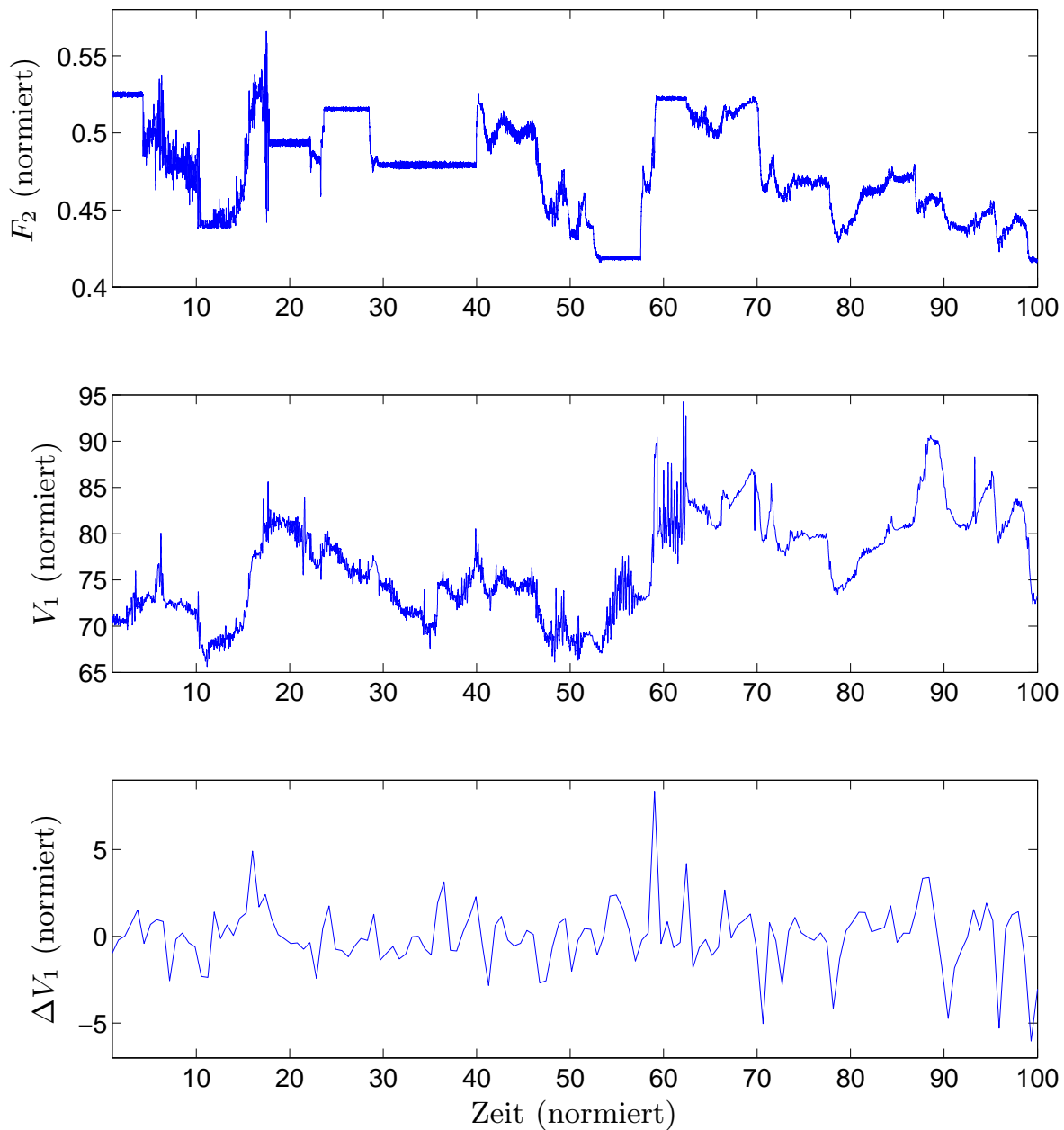


Abbildung 8.7: Gemessene Verläufe des Rückflusses F_2 und der Ventilstellung im Zufluss V_1 .

durch Optimierung der Parameter unterstreichen die Eignung DRFS zur Nachbildung nichtlinearer Prozesse.

Allerdings kann das DRFS aufgrund seiner hauptsächlich qualitativen Charakteristik nur die langsamen Anteile der Kolonnendynamik nachbilden. Schnellere Dynamiken werden dagegen nicht modelliert. Möchte man auch die schnelle Dynamik erfassen, so sind zeitkontinuierliche bzw. zeitdiskrete rekurrente Fuzzy-Systeme ohne Rückkopplungskorrespondenz besser

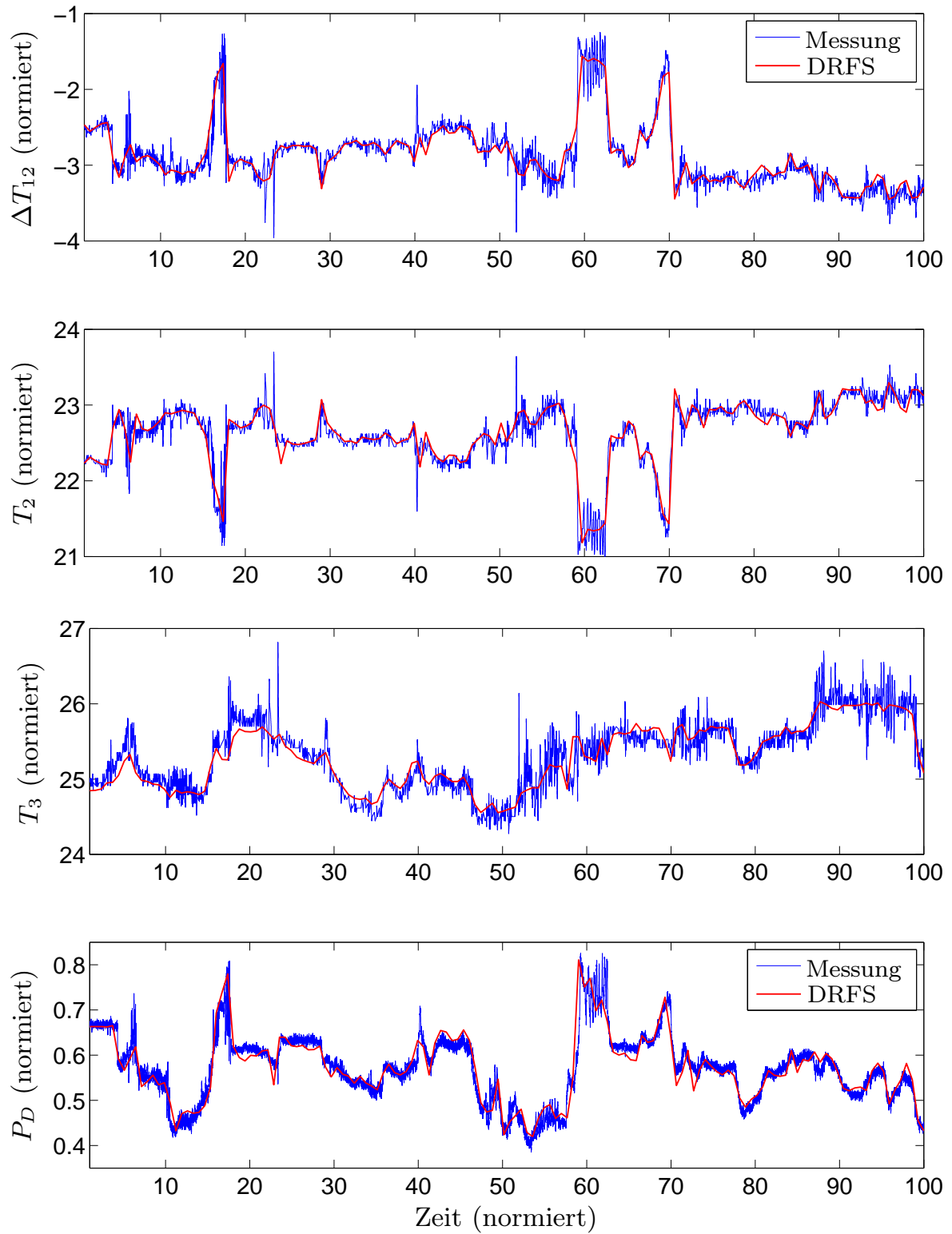


Abbildung 8.8: Vergleich der gemessenen und mit dem DRFS modellierten Verläufe für die Zustandsgrößen ΔT_{12} , T_2 , T_3 und P_D für den gesamten Datensatz (Einschrittprediktion).

geeignet. Allerdings ergeben sich dann sehr große Regelbasen, die kaum mehr interpretierbar sind.

8.2 Fehlerdiagnose an einer hydraulischen Servoachse

In diesem Abschnitt soll das in Kapitel 7 vorgestellte Konzept zur Fehlerdiagnose mit rekurrenten Fuzzy-Systemen an einer elektrohydraulischen Servoachse mit Duplex-Ventilsystem² auf seine Praxistauglichkeit hin untersucht werden. Die Anordnung besteht aus einem Hydraulikzylinder, der über zwei Standard-Proportionalwegeventile aktuiert wird. Aufgrund der redundanten Auslegung durch die parallele Anordnung der Ventile bietet das System Möglichkeiten zur Rekonfiguration der Servoachse beim Auftreten von Fehlern. Hierzu ist eine zuverlässige Diagnose der Fehler besonders wichtig.

Die Einsatzmöglichkeiten solcher redundant ausgelegten elektrohydraulischen Servoantriebe sind breit gestreut. Neben klassischen Anwendungen in sicherheitsrelevanten Systemen, wie Fahrwerken und Landeklappen in Flugzeugen, werden sie zunehmend auch in industriellen Systemen, wie beispielsweise Werkzeugmaschinen eingesetzt.

Da es sich bei dem Duplex-Ventilsystem um ein elektrohydraulisches System handelt, sind die physikalischen Zusammenhänge mit Ausnahme einiger Parameter bekannt. Entsprechend erfolgt die Modellbildung auf Basis von physikalischen Gleichungen. Sie ist in [18] detailliert dargestellt. Auf eine Modellierung des Systems mit RFS wurde deshalb verzichtet. Vielmehr richtet sich der Fokus der Untersuchungen in diesem Abschnitt auf die *Fehlerisolation* mit zeitdiskreten RFS mit Rückkopplungskorrespondenz, wie sie in Abschnitt 7.3 vorgestellt wurde.

Im Folgenden soll zunächst ein Überblick über die betrachtete Servoachse gegeben werden. Da die Residuen modellgestützt gebildet werden, wird hierbei auch kurz auf die physikalische Modellbildung der einzelnen Komponenten eingegangen. Daran anschließend werden die zu untersuchenden Fehlerfälle definiert. Abschließend wird die Diagnose dieser Fehlerfälle mithilfe von DRFS präsentiert und die erreichten Ergebnisse diskutiert.

²Der Versuchsstand mit der Servoachse wurde in der Forschungsgruppe Regelungstechnik und Prozessautomatisierung des Instituts für Automatisierungstechnik von Mark Beck aufgebaut. Für die Möglichkeit, den Versuchsstand zu nutzen, und die gute Zusammenarbeit möchte ich mich herzlich bedanken.

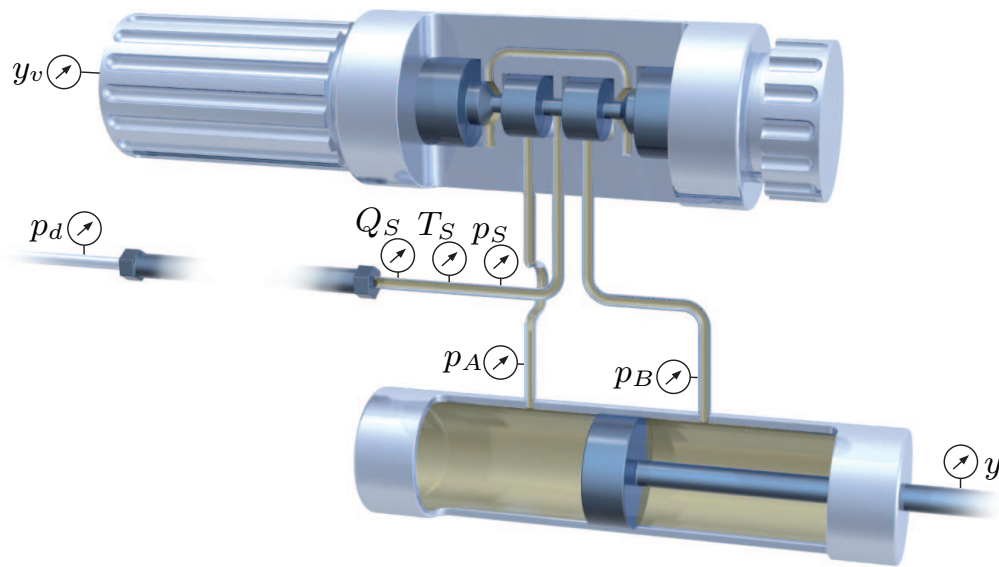


Abbildung 8.9: Schematischer Aufbau der hydraulischen Servoachse.

8.2.1 Systembeschreibung

Die betrachtete elektrohydraulische Servoachse besteht aus zwei wesentlichen Komponenten. Dies sind zum einen die beiden identischen 4/3 Proportionalwegeventile (Bosch Rexroth 4WRE) und zum anderen der Hydraulikzylinder (vgl. Abbildung 8.9). Ziel des Systems ist es, eine an den Zylinder gekoppelte Last kontinuierlich zu bewegen und möglichst genau zu positionieren. Da beide Ventile baugleich sind, kann auch die Fehlerdiagnose für beide Ventile gleich ausgelegt werden. Entsprechend wird im Folgenden nur noch ein Ventil betrachtet. Den schematischen Aufbau zeigt Abbildung 8.9 [18, 139].

Das Ventil kann in einen elektromagnetischen und einen hydraulischen Teil unterteilt werden. Der elektromagnetische Teil besteht aus zwei Gleichstrommagneten in Differenzanordnung [18]. Durch entsprechende Bestromung der Magneten kann der Ventilschieber kontinuierlich bewegt werden. Die Position des Ventilschiebers stellt gleichzeitig die Eingangsgröße des hydraulischen Teils dar. Über die Stellung des Ventilschiebers lassen sich die hydraulischen Leitwerte und damit, abhängig vom Lastdruck, die Volumenströme an den vier Steuerkanten des Ventils kontinuierlich variieren. Die Volumenströme werden dann in die beiden Zylinderkammern geleitet, wodurch sich am Zylinderkolben ein Differenzdruck aufbaut. Dieser Differenzdruck erzeugt schließlich die Kraft zur Positionierung der Last am

Zylinderkolben. Der Versorgungsdruck des hydraulischen Teils wird durch eine Pumpe bereitgestellt. Als Last ist ein Masse-Feder-Dämpfer-System angekoppelt. Außerdem werden zur Positionierung der Last zwei Regler verwendet. Hierbei wird im äußeren Regelkreis die Lastposition geregelt. Dieser wird um einen unterlagerten Regler zur Einstellung der Ventilschieberposition ergänzt. Als Regler kommen jeweils PID-Regler zum Einsatz.

Zur Untersuchung des Systems, insbesondere zur Regelung und Fehlerdiagnose, werden verschiedene Größen gemessen. Dies sind die Ventilschieberposition y_v als Eingangsgröße des hydraulischen Teils, die Kolbenposition y , der von der Pumpe gelieferte Druck p_d , der im Betrieb annähernd konstant gehalten wird, der dem System zur Verfügung gestellte Versorgungsdruck p_s sowie die Drücke in den beiden Zylinderkammern p_A und p_B .

Mithilfe der zur Verfügung stehenden Messgrößen wurden in [19, 20, 21] modellbasiert Residuen erzeugt. Die ersten beiden Residuen werden mithilfe der Kammerdrücke generiert. Für den modellierten Druckaufbau in den beiden Zylinderkammern \hat{p}_A und \hat{p}_B gilt

$$\hat{p}_A = \frac{E(\hat{Q}_{PA} - \hat{Q}_{AT} - G_{AB}(p_A - p_B) - A_A \dot{y})}{V_{0A} + A_A y}, \quad (8.1)$$

$$\hat{p}_B = \frac{E(\hat{Q}_{PB} - \hat{Q}_{BT} + G_{AB}(p_A - p_B) + A_B \dot{y})}{V_{0B} - A_B y}, \quad (8.2)$$

wobei \hat{Q}_{PA} bzw. \hat{Q}_{PB} den modellierten Zufluss in die Kammer A bzw. B, \hat{Q}_{AT} bzw. \hat{Q}_{BT} den modellierten Abfluss aus der Kammer A bzw. B und y die Kolbenposition beschreibt. Außerdem sind V_{0*} bzw. A_* die Totvolumina bzw. aktive Kolbenflächen von Kammer A bzw. B, E ist das Elastizitätsmodul und G_{AB} der laminare Leakagekoeffizient. Letzterer wird in den Modellgleichungen im Folgenden zu null angenommen. Dies erlaubt später die Erkennung von interner Leakage. Die Modellierung der Volumenströme geschieht über ein semiphysikalisches Modell der Steuerkanten, das in [20] ausführlich beschrieben ist. Die Größe der Volumenströme \hat{Q}_A und \hat{Q}_B hängt hierbei nichtlinear von der Ventilschieberposition y_v , der Ventilüberdeckung s_0 , dem Versorgungsdruck p_s und den Kammerdrücken p_A und p_B ab, d.h., $\hat{Q}_* = \hat{Q}_{P*} - \hat{Q}_{*T} = f(y_v, p_s, p_A, p_B, s_0)$. Die Parametrierung des gesamten Modells ist in Anhang B angegeben.

Aus der Differenz der mithilfe eines Zustandsvariablenfilters [79] aus den gemessenen Drücken berechneten Druckableitungen \dot{p}_A und \dot{p}_B und den

mithilfe von Gl. (8.1) - (8.2) modellierten Druckableitungen \dot{p}_A und \dot{p}_B ergeben sich nun die beiden Residuen

$$\begin{aligned} r_{p_A} &= \dot{p}_A - \hat{\dot{p}}_A, \\ r_{p_B} &= \dot{p}_B - \hat{\dot{p}}_B, \end{aligned}$$

in denen die Kolbengeschwindigkeit \dot{y} aus der Kolbenposition y wiederum mithilfe eines Zustandsvariablenfilters gebildet wird.

Neben den angegebenen modellbasierten Residuen wird zusätzlich als signalbasiertes Residuum das Stellsignal y_v überwacht. Falls dieses nicht im Normalbereich, sondern ober- oder unterhalb zweier gegebener Grenzwerte liegt, wird dieses Residuum auf eins gesetzt, d.h.,

$$r_{\text{sat}} = \begin{cases} 0, & \text{wenn } y_{v, \min} < y_v < y_{v, \max} \\ 1, & \text{sonst.} \end{cases}$$

Das Residuum r_{sat} eignet sich insbesondere zur Erkennung der Zylinderblockade.

Die drei Residuen r_{p_A} , r_{p_B} und r_{sat} weisen im fehlerfreien Fall nur kleine Ausschläge auf, die auf Messrauschen und Modellungenauigkeiten zurückzuführen sind. Wie sich die Residuen im Fehlerfall verhalten, wird im folgenden Abschnitt nach der Definition der auftretenden Fehler gezeigt.

8.2.2 Beschreibung der Fehlerfälle

Nach der Vorstellung des Systems und der modellbasierten Residuengenerierung sollen nun die möglichen Fehlerfälle an der Servoachse definiert und beschrieben werden. Hierbei konzentrieren sich die Betrachtungen auf den hydraulischen Teil der Anlage.

Grundsätzlich lassen sich Fehler nach verschiedenen Kriterien charakterisieren. Dies ist zum einen das Zeitverhalten der Fehler und zum anderen die Art, wie die Fehler modelliert werden. Das Zeitverhalten kann gemäß [80] in abrupte, schleichende und sporadisch auftretende Fehler eingeteilt werden. Abrupte Fehler treten zu einem bestimmten Zeitpunkt auf und bleiben danach bis zur Behebung existent, wie beispielsweise bei mechanischer Blockade. Schleichende Fehler entwickeln sich langsam, wobei die Größe kontinuierlich zunimmt. Typische Beispiele sind Alterungserscheinungen, Sensordrift oder Verstopfungen. Sporadische Fehler treten kurz auf und verschwinden dann wieder, wie beispielsweise in Wackelkontakten.

Bei der Fehlermodellierung unterscheidet man zwischen multiplikativen und additiven Fehlern [80]. Zur ersten Gruppe gehören Veränderungen in

Streckenparametern wie Durchflusskoeffizienten und Ähnliches. Sie werden durch $k = k_0(1 + \Delta f)$ beschrieben, wobei k den veränderlichen Parameter und Δf die Fehlergröße beschreibt. Zur zweiten Gruppe gehören beispielsweise Offsetfehler in Sensoren, die durch $k = k_0 + \Delta f$, also einer additiven Fehlereinwirkung, beschrieben werden.

Die an der hydraulischen Servoachse definierten Fehler³, ihr angenommenes Zeitverhalten und ihre Modellierung sind in Tabelle 8.3 dargestellt. Die betrachteten Fehler in der Sensorik beziehen sich vor allem auf die

Tabelle 8.3: Übersicht über die betrachteten Fehlerfälle an der Servoachse.

	Fehler	Zeitverhalten	Modellierung
φ_1	Sensordrift (Druck p_A)	schleichend	additiv
φ_2	Sensordrift (Druck p_B)	schleichend	additiv
φ_3	Sensordrift (Druck p_S)	schleichend	additiv
φ_4	Positionssignal hängt	abrupt	$y = konst., \dot{y} = 0$
φ_5	Interne Leckage G_{AB}	schleichend	multiplikativ
φ_6	Niedrigeres E-Modul E_0	abrupt	multiplikativ
φ_7	Zylinderblockade	abrupt	$y = konst., \dot{y} = 0$
φ_8	Offset im Ventilschieber	schleichend	additiv
φ_9	Ventilschieber blockiert	abrupt	$y_v = konst., \dot{y}_v = 0$

Drucksensoren in den beiden Zylinderkammern sowie auf den Drucksensor des Versorgungsdrucks. Ursachen für solche Driftfehler, die als schleichende Fehler modelliert werden, liegen z.B. in Alterungseffekten oder falscher Kalibrierung der Sensoren. Wie bei Sensorfehlern üblich, werden sie als additive Fehler modelliert. Außerdem wird ein Fehler im Positionssensor des Zylinders betrachtet. Ein Szenario für das Auftreten eines solchen Fehlers ist eine kurzfristig blockierende Mechanik, die zum „Einfrieren“ des Sensorsignals führt. Modelliert wird dieser abrupte Fehler durch ein ab dem Fehlerzeitpunkt konstantes Sensorsignal.

Des Weiteren werden einige Prozessfehler betrachtet. Als parametrischer und damit multiplikativer Fehler wird eine erhöhte interne Leckage im Zylinder angenommen. Dies führt zu einem zusätzlichen Volumenstrom zwi-

³Eine Übersicht über Fehlerstatistiken für hydraulische Servoachsen findet sich in [139].

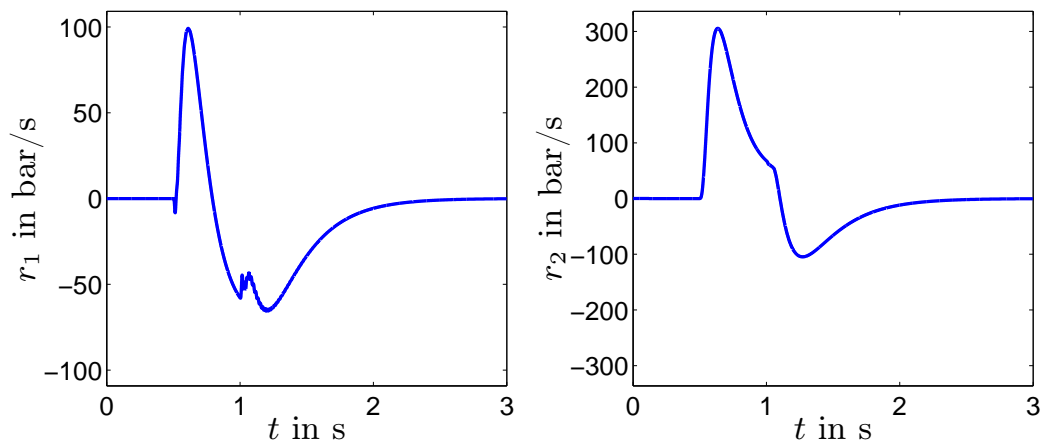


Abbildung 8.10: Dynamische Residuenverläufe r_1 und r_2 bei einer sprunghaften Verringerung des Elastizitätsmoduls bei $t = 0,5$ s.

schen den Zylinderkammern und wird durch Abnutzung der Dichtung hervorgerufen. Ein weiterer parametrischer Fehler ist die Verringerung des Elastizitätsmoduls des Öls, die durch Lufteinschlüsse im Öl entsteht, z.B. weil aufgrund eines Defekts die versorgende Pumpe teilweise Luft fördert. Außerdem wird die Blockade des Zylinders als Fehlerfall definiert. Ähnlich wie beim Fehler des Positionssignals wird wiederum die Position ab dem Fehlerzeitpunkt als konstant angenommen. Als letzte Fehlerquelle wird der Ventilschieber betrachtet. Hier wird sowohl eine vollständige Blockade als auch ein Offset im Ventilschieber, der durch eine fehlerhafte Kalibration entsteht, berücksichtigt. Erstere wird wiederum durch ein konstantes Positionssignal des Ventilschiebers modelliert, letztere stellt einen additiven Fehler dar.

Die Auswirkung dieser Fehler auf die Residuen soll nun zunächst für die Einzelfehler untersucht werden. Fehlersymptomtabelle 8.4 zeigt, welche Residuen auf den betrachteten Fehlerfall sensitiv sind und welche Ausschläge (positiv oder negativ) erzeugt werden. Aus der Tabelle ist ersichtlich, dass alle Einzelfehler eindeutig isoliert werden können, da sich die Reihen jeweils an mindestens einer Stelle unterscheiden. Außerdem treten einige spezielle Erscheinungen auf. So führt das „Einfrieren“ des Positionssensors am Zylinder, das eine Nichtlinearität in den Positionsregelkreis einbringt, zu einem Grenzzyklus im Druckresiduum von Kammer B. Eine Verringerung des Elastizitätsmoduls durch Lufteinschlüsse resultiert in einem dynamischen Residuenverlauf, den Abbildung 8.10 zeigt. Bei Auftreten des Fehlers schlägt das Residuum zunächst positiv aus, wird dann aber nach einem negativen Überschwingen wieder zu null geregelt.

Tabelle 8.4: Fehlersymptomtabelle des hydraulischen Teilsystems. Hierbei ist + ein positiver, – ein negativer und o kein Ausschlag. Die Abkürzung „gz“ bedeutet Grenzzyklus, „dyn“ steht für dynamischer Residuenverlauf. Bei Fehler φ_9 kann die Blockade sowohl in positiver als auch negativer Position bezogen auf den Referenzpunkt auftreten.

Fehler	r_{pA}	r_{pB}	r_{sat}
φ_0 : fehlerfrei	o	o	o
φ_1 : Sensordrift (Druck in Kammer A)	+	o	o
φ_2 : Sensordrift (Druck in Kammer B)	o	+	o
φ_3 : Sensordrift (Versorgungsdruck)	-	-	o
φ_4 : Positionssignal ausgefallen	-	gz	+
φ_5 : Interne Leckage G_{AB}	-	+	o
φ_6 : Niedrigeres E-Modul E_0	dyn	dyn	o
φ_7 : Zylinderblockade	o	o	+/-
φ_8 : Offset im Ventilschieber	+	-	o
φ_9 : Ventilschieber blockiert (+/-)	-/+	+/-	+/-

Zusätzlich zu den Einzelfehlern sollen im Folgenden auch Doppelfehler untersucht werden, wobei davon ausgegangen wird, dass die Fehler nacheinander auftreten. Hierzu wird angenommen, dass die Zeit zwischen dem Auftreten der beiden Fehler zur Isolation des ersten Fehlers ausreichend ist. Diese Annahme ist dadurch gerechtfertigt, dass die Zeitdauer bis zur Fehlerisolation im Anwendungsbeispiel maximal etwa drei Sekunden beträgt. Die Fehlersymptomtabelle für alle Doppelfehlerfälle ist in Anhang B.2 angegeben. Hierbei werden nur die in der Praxis relevanten Doppelfehlerfälle betrachtet. So wird die Blockade von Zylinder und Ventilschieber sowie ein hängendes Positionssignal nur als zweiter auftretender Fehler berücksichtigt. Treten diese Fehler zuerst auf, so muss das System aufgrund der Schwere des Fehlers direkt abgeschaltet werden, sodass die Erkennung eines zweiten Fehlers weder sinnvoll noch notwendig ist.

8.2.3 Fehlerdiagnose

Auf Basis der definierten Fehlerfälle und Residuen sowie der daraus resultierenden Fehlersymptomtabelle kann nun mit Hilfe der Methodik aus Abschnitt 7.3 ein zeitdiskretes RFS zur Isolation der Einzel- und Doppelfehler ausgelegt werden. Hierbei wird der Übersichtlichkeit halber nur ein Ventil und der Zylinder betrachtet. Die Erweiterung um das zweite Ventil ist aber einfach möglich. Bei der Auslegung des zeitdiskreten RFS werden binäre Zugehörigkeitsfunktionen für die Fehlerzustände verwendet, die sich als zweckmäßiger für diese Anwendung herausgestellt haben. Insbesondere ist die Auslegung dahingehend einfacher, dass die Dynamik des DRFS aufgrund der binären Zugehörigkeitsfunktionen einfacher nachvollziehbar ist. Es ist aber zu beachten, dass die Eingangsgrößen weiterhin fuzzifiziert werden. Außerdem wurde weitestgehend auf die Verwendung von Verstärkerregeln verzichtet (vgl. Abschnitt 7.3.2). Allerdings müssen variierende Führungsgrößen berücksichtigt werden, da das Verhalten der Residuen in einigen Fehlerfällen stark von der Führungsgröße abhängig ist. Entsprechend wird die Führungsgröße als zusätzliche Eingangsgröße ins DRFS geführt, wobei sie mithilfe der drei Zugehörigkeitsfunktionen „positiv“, „null“ und „negativ“ fuzzifiziert wird. Es ergibt sich damit als Eingangsvektor des RFS $(\varphi, r_{pA}, r_{pB}, r_{sat}, y_v)$

Im Folgenden wird nun zunächst das Gesamtergebnis der Fehlerisolation für Einzel- und Doppelfehler diskutiert. Anschließend werden einige repräsentative Fehlerszenarien betrachtet. Da einige Fehlerfälle, wie beispielsweise Luft einschlüsse, am Versuchsstand nicht implementiert werden können, werden diese Szenarien nur mithilfe des nichtlinearen Simulationsmodells untersucht. Alle anderen Szenarien wurden am Simulationsmodell und am Teststand untersucht.

Ergebnis für Einzelfehler

Zunächst werden anhand von Fehlersymptomtabelle 8.4 die Einzelfehler betrachtet. Offenbar sind die Sensorfehler $\varphi_1 - \varphi_3$ unterscheidbar und können somit bereits durch ein SFS isoliert werden. Ähnliches gilt für die Zylinderblockade φ_7 , die eindeutig bei alleinigem Auftreten von Residuum r_{sat} erkannt werden kann. Problematisch bei der Verwendung von SFS sind die dynamischen Residuenverläufe bei Fehler φ_4 und φ_6 . Im ersten Fall weisen bei fortschreitendem Fehler die Residuen von Fehler φ_4 und Fehler φ_9 zeitweise die gleiche Charakteristik $-++$ auf. Zur Unterscheidung ist ein RFS notwendig, um das durch den Grenzzyklus entstehende Muster isolieren zu

können. Alternativ kann der Grenzyklus direkt mithilfe eines Filters detektiert werden, dessen Ausgang als zusätzliches Residuum verwendet wird. Allerdings erhöht sich damit die Zahl der Residuen um eins. Im zweiten Fall, dem verringerten Elastizitätsmodul, ist dies allerdings nicht möglich. Hier stimmen die Residuen von Fehler φ_6 zeitweise mit den Residuen von Fehler φ_5 und Fehler φ_8 überein. Dieser Fehler ist nur mithilfe des RFS isolierbar, da dieses auch Residuenmuster, wie in Abbildung 8.10 gezeigt, abbilden kann. Insgesamt sind also mithilfe des RFS alle Einzelfehler isolierbar. Das SFS kann dagegen einige Fehler nicht eindeutig isolieren.

Ergebnis für Doppelfehler

Zusätzlich zu den Einzelfehlern sollen auch die zuvor definierten Doppelfehler betrachtet werden. Die zugehörige Symptomtabelle ist in Tabelle B.7 im Anhang gegeben. Die daraus folgenden Ergebnisse zur Isolierbarkeit der Fehler sind in Tabelle 8.5 dargestellt. Insgesamt ergeben sich drei verschiedene Klassen:

- Grün hinterlegt: Diese Fehler sind mit dem DRFS eindeutig isolierbar. Diese Gruppe enthält alle Einzelfehlerfälle sowie solche Doppelfehler, bei denen als zweiter Fehler eine Zylinder- bzw. Ventilschieberblockade (Fehler φ_7 und φ_9) sowie ein ausgefallenes Positionssignal (Fehler φ_4) vorliegt. Dies liegt insbesondere in der dominierenden Wirkung dieser Fehler gegenüber dem ersten Fehler begründet. Des Weiteren sind die Doppelfehler mit E-Modul-Fehler größtenteils eindeutig isolierbar. Dies resultiert aus den dynamischen Residuenverläufen, die sich eindeutig von anderen Doppelfehlern unterscheiden.
- Orange hinterlegt: Bei diesen Fehler hängt die Isolierbarkeit von den Fehlergrößen ab. Der zweite auftretende Fehler muss so groß sein, dass sich die Residuen gegenüber dem ersten Fehler ändern. Ist dies nicht der Fall, so können die Fehler nicht isoliert werden. Ist der Fehlereinfluss groß genug, so lassen sich nochmal zwei Klassen unterscheiden. Zum einen existieren Doppelfehler, die dann eindeutig isolierbar sind. Dies gilt z.B. für Fehler φ_{13} . Zum anderen existieren Doppelfehler, deren Residuen nicht von der Fehlergröße abhängen, die aber dieselbe Residuencharakteristik wie größenabhängige Doppelfehler haben. In diesem Fall lassen sich nur Fehlergruppen isolieren. Ein Beispiel hierzu ist Fehler φ_{12} , der die größenunabhängige Charakteristik $++0$ besitzt. Allerdings kann auch Fehler φ_{15} bei entsprechender Fehler-

Tabelle 8.5: Übersicht über die Isolierbarkeit der Fehlerfälle am Ventilsystem bei Verwendung des DRFS. Die grün gekennzeichneten Fälle sind eindeutig isolierbar, die rot gekennzeichneten Fälle sind nicht isolierbar. Bei den orange gekennzeichneten Fällen hängt die Isolierbarkeit von den Fehlergrößen ab. Teilweise lassen sich dann nur Fehlergruppen isolieren.

	2. Fehler								
	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7	φ_8	φ_9
1. Fehler									
φ_1	φ_1	φ_{12}	φ_{13}	φ_{14}	φ_{15}	φ_{16}	φ_{17}	φ_{18}	φ_{19}
φ_2	φ_{21}	φ_2	φ_{23}	φ_{24}	φ_{25}	φ_{26}	φ_{27}	φ_{28}	φ_{29}
φ_3	φ_{31}	φ_{32}	φ_3	φ_{34}	φ_{35}	φ_{36}	φ_{37}	φ_{38}	φ_{39}
φ_5	φ_{51}	φ_{52}	φ_{53}	φ_{54}	φ_5	φ_{56}	φ_{57}	φ_{58}	φ_{59}
φ_6	φ_{61}	φ_{62}	φ_{63}	φ_{64}	φ_{65}	φ_6	φ_{67}	φ_{68}	φ_{69}
φ_8	φ_{81}	φ_{82}	φ_{83}	φ_{84}	φ_{85}	φ_{86}	φ_{87}	φ_8	φ_{89}

größe diese Charakteristik aufweisen. Tritt also diese Charakteristik auf, so kann man nur auf die Fehlergruppe $(\varphi_{12}, \varphi_{15})$ schließen.

- Rot hinterlegt: Diese Fehler sind mit dem DRFS nicht isolierbar. Hierbei handelt es sich um Doppelfehler, bei denen der zweite Fehler ein E-Modul-Fehler ist. Die durch diesen Fehler entstehende Residuedynamik ist aufgrund der Einflüsse der Erstfehler nicht erkennbar, sodass der zweite Fehler nicht erkannt werden kann. Außerdem ist Fehler φ_{81} nicht isolierbar, da sich die Residuen von φ_8 und φ_{81} nicht unterscheiden.

Zusammenfassend sind also mithilfe des DRFS die neun Einzelfehler sowie 23 Doppelfehler und damit 56% der Fehler eindeutig isolierbar. Bei 21 Doppelfehlern (37% der Fälle) hängt die Isolierbarkeit von der Größe des zweiten auftretenden Fehlers ab. Ist dessen Wirkung auf die Residuen gegenüber dem ersten Fehler merkbar groß, so sind die Fehler eindeutig oder in einer Fehlergruppe isolierbar. Ansonsten ist eine Isolation nicht möglich. Vier Doppelfehler (7%) sind überhaupt nicht isolierbar. Mit einem SFS ist dagegen kein Doppelfehler eindeutig isolierbar. Es ist lediglich möglich, Fehlergruppen zu isolieren. Allerdings sind diese Gruppen in den meisten Fällen sehr groß, enthalten also viele Fehler, was zur praktischen Nutzung nicht mehr sinnvoll ist. Insgesamt ergibt sich also durch die Verwendung von DRFS eine deutliche Verbesserung der Fehlerisolation.

Szenario 1: Verringerung des E-Moduls (Simulation)

Nach der Vorstellung des Gesamtergebnisses soll die Methodik nun an einigen ausgewählten Szenarien illustriert werden. Als erstes Szenario wird das Auftreten von Fehler φ_6 , d.h. eine Verringerung des Elastizitätsmoduls um 50%⁴, untersucht. Dieser Fall ist insbesondere deshalb interessant, weil es sich um einen Fehler mit dynamischem Residuenverlauf handelt. Wie ein solcher Verlauf durch das DRFS berücksichtigt werden kann, soll nun erläutert werden. Abbildung 8.11 zeigt hierzu im oberen Teil die Residuenverläufe von r_{p_A} und r_{p_B} sowie die Schwellen, die durch die Kernpositionen beschrieben werden. Das Residuum r_{sat} ist nicht sensitiv auf diesen Fehlerfall ($r_{sat} = 0$) und ist deshalb nicht abgebildet. Wie zu sehen, schlagen nach Auftreten des Fehlers beide Residuen zunächst positiv aus. Es ergibt sich also zunächst die Residuencharakteristik $++0$. Anschließend fallen beide Residuen zeitversetzt ab, sodass sich als Reihenfolge der linguistischen Werte im weiteren Verlauf $0+0 \rightarrow -+0 \rightarrow -00 \rightarrow 0-0 \rightarrow 000$ ergibt. Diese Reihenfolge der linguistischen Werte der Residuen kann, ähnlich wie in Abschnitt 7.3.2 beschrieben, in Form eines linguistischen Automaten dargestellt werden. Dieser definiert schließlich die Regelbasis des DRFS.

Die Ergebnisse der Fehlerisolation sind in Abbildung 8.11 unten zu sehen. Zunächst wird kurzzeitig mit einer maximalen Zugehörigkeit von 0,15 Fehler φ_2 angezeigt. Dies ist darauf zurückzuführen, dass die beiden Residuen leicht zeitversetzt auf den Fehler reagieren, sodass sich zunächst eine geringe Zugehörigkeit zum Muster $0+0$ ergibt, was wiederum auf Fehler φ_2 hindeutet (vgl. Tabelle 8.4). Anschließend durchlaufen die Residuen das oben beschriebene Muster aufeinander folgender linguistischer Werte. Sobald der letzte Teil des Musters mit einer bestimmten Zugehörigkeit erfüllt ist, wird der Fehler φ_6 korrekt isoliert.

Es sei angemerkt, dass bei etwas veränderter Auslegung des DRFS der Fehler bereits während des Durchlaufens des Musters mit ansteigender Zugehörigkeit angezeigt werden kann. Dies ist insbesondere dann wünschenswert, wenn die Residuendynamik mit einer langsamen Zeitkonstanten reagiert. In diesem Anwendungsfall durchlaufen die Residuen das Muster allerdings innerhalb einer Sekunde. Um unnötige Fehllalarme zu vermeiden, wird erst nach dem Durchlauf des Musters der Fehler angezeigt.

⁴Dies entspricht bei 80 bar etwa einem Volumenverhältnis von Luft zu ÖL von 0,6 [47].

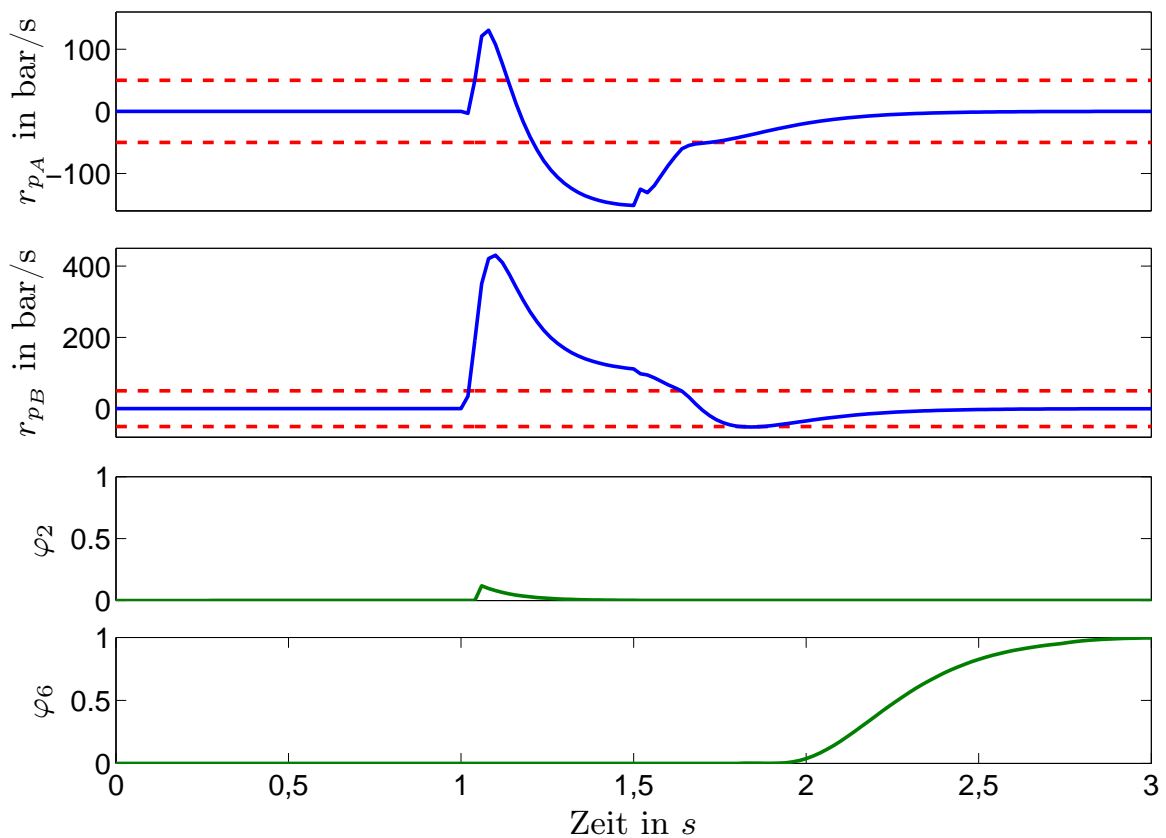


Abbildung 8.11: Residuenverläufe r_{p_A} und r_{p_B} (durchgezogen) mit Kernpositionen der Residuen (gestrichelt) sowie Ergebnisse der Fehlerisolation (unten) bei einer Verringerung des Elastizitätsmoduls um 50%. Zunächst wird kurzzeitig mit geringer Zugehörigkeit Fehler φ_2 erkannt. Nach Erkennung des Residuenmusters wird der tatsächliche Fehler φ_6 isoliert.

Szenario 2: Doppelfehler mit Drift im Drucksensor und anschließender Verringerung des E-Moduls (Simulation)

Als zweites Szenario wird ein Doppelfehler bestehend aus einem Sensorfehler und einem Fehler mit dynamischem Residuenverlauf betrachtet. Zunächst tritt ab dem Zeitpunkt $t = 1\text{s}$ ein Drift im Drucksensor der Kammer B auf. Zum Zeitpunkt $t = 5\text{s}$ verringert sich zusätzlich das Elastizitätsmodul um 50%. Die Residuenverläufe und Ergebnisse der Fehlerisolation sind in Abbildung 8.12 dargestellt. Das DRFS erkennt beide Fehler φ_2 und φ_6 jeweils kurz nach ihrem Auftreten, wobei der Fehler φ_6 wie im vorigen Szenario erst nach Durchlaufen der Residuenmuster isoliert wird.

Darüber hinaus ist der Verlauf der Ventilschieberposition y_v , abgebildet. Letztere wird, wie bereits erwähnt, als zusätzliche Eingangsgröße ins DRFS geführt, da sie den Verlauf der Residuen deutlich beeinflusst. Dies ist in Abbildung 8.12 beim Abfahren der Rampe ab dem Zeitpunkt $t = 7\text{s}$ zu

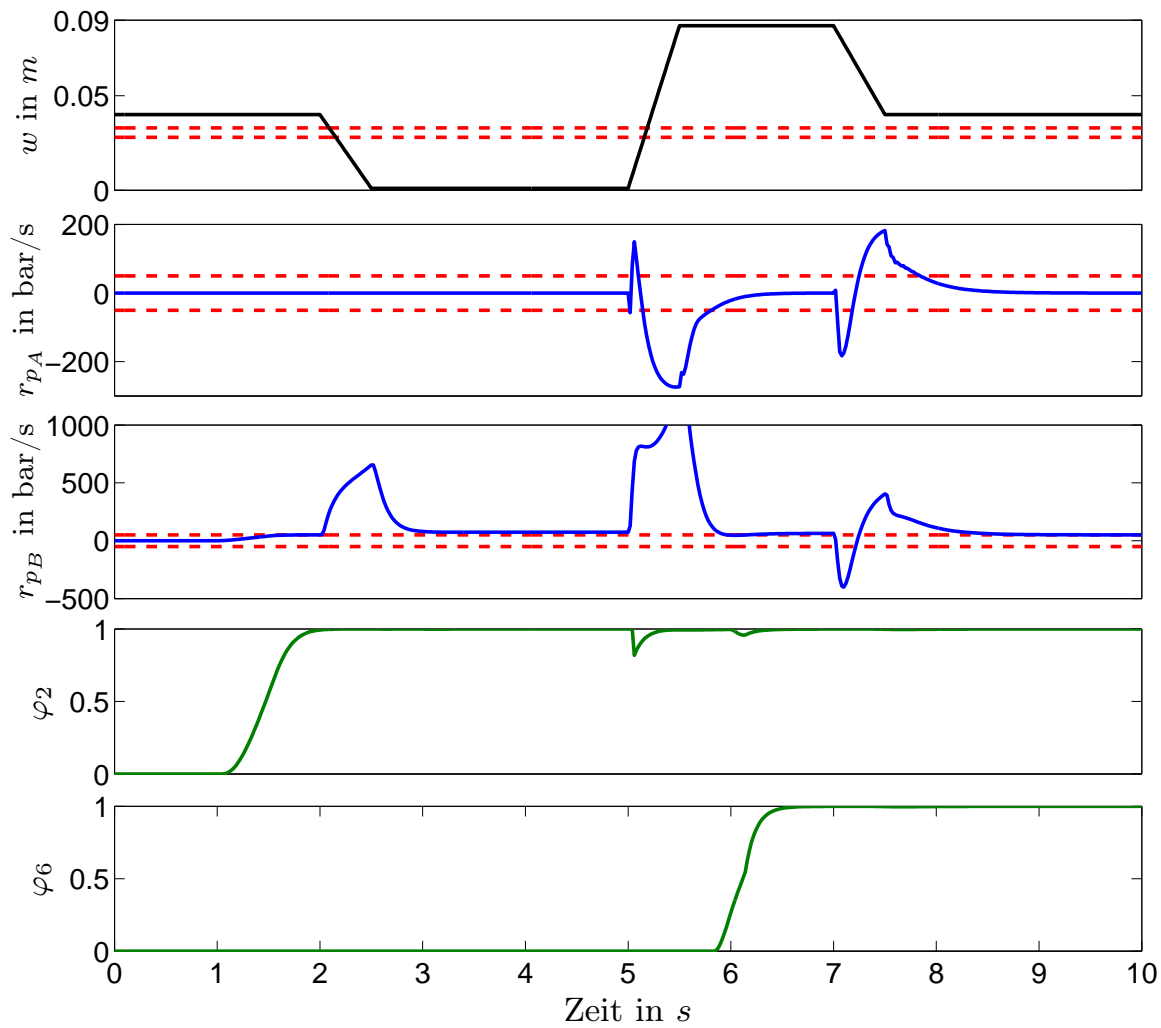


Abbildung 8.12: Verlauf der Führungsgröße w und der Residuenverläufe r_{pA} und r_{pB} (durchgezogen) mit Kernpositionen (gestrichelt) sowie Ergebnisse der Fehlerisolation (unten) bei Fehlerszenario 2. Beide Fehler φ_2 und φ_6 werden kurz nach dem Auftreten korrekt isoliert.

erkennen, bei dem die Residuen aufgrund des verringerten Elastizitätsmoduls wiederum ausschlagen. Durch die Berücksichtigung der Führungsgröße kann das DRFS nun so ausgelegt werden, dass Residuenausschläge beim Abfahren der Rampe als Indiz für den Fehler gewertet werden. Ohne Berücksichtigung der Führungsgröße können solche Ausschläge dagegen die Fehlerzustände zurücksetzen.

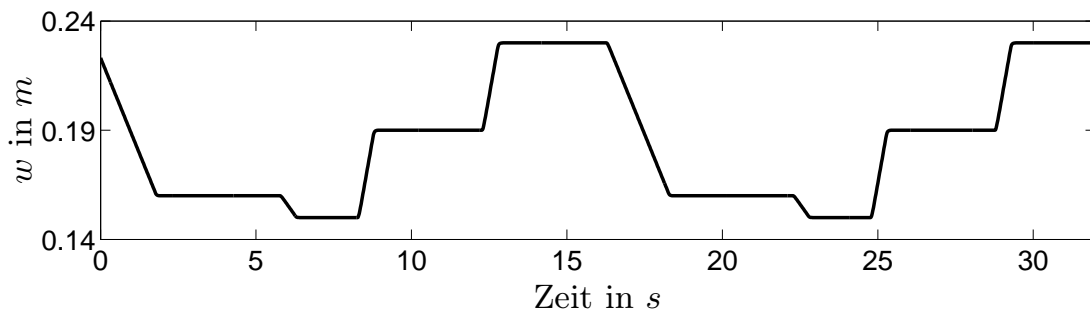


Abbildung 8.13: Führungssignal für die Messungen am Prüfstand.

Szenario 3: Doppelfehler mit Drift im Drucksensor und anschließender Ventilschieberblockade (Messung)

Nach den zuvor betrachteten Szenarien, die anhand des Simulationsmodells betrachtet wurden, sollen nun auch zwei direkt am System eingetragene Doppelfehlerszenarien gezeigt werden. Hierbei wird das System mit dem in Abbildung 8.13 gezeigten Führungssignal w beaufschlagt. Da im Modell die Leckage vernachlässigt wurde, am realen System aber immer leichte Leckageverluste entstehen, weisen beide Druckresiduen r_{pA} und r_{pB} einen Offset auf. Deshalb sind die Zugehörigkeitsfunktionen nicht mehr symmetrisch um null, sondern symmetrisch um den Offset platziert. Außerdem werden nun trapezförmige Zugehörigkeitsfunktionen für den linguistischen Wert „null“ verwendet, um die Isolation gegenüber Messrauschen und anderen Störeinflüssen unempfindlicher zu machen.

Im ersten betrachteten Szenario tritt zum Zeitpunkt $t = 9\text{s}$ ein Drift im Drucksensor in Zylinderkammer A auf. Zusätzlich blockiert zum Zeitpunkt $t = 18,5\text{s}$ der Ventilschieber. Abbildung 8.14 enthält die sich ergebenden Verläufe der Residuen r_{pA} und r_{pB} sowie die angezeigten Fehler. Bei Auftreten des ersten Fehlers schlägt zunächst nur Residuum r_{pA} in positiver Richtung aus. Allerdings ist der Ausschlag aufgrund des geringen Drifts relativ gering und liegt im Bereich der Kernpositionen des Residuums. Entsprechend erkennt das DRFS den Fehler φ_1 zunächst nur mit einer bestimmten Zugehörigkeit. Durch die Rückführung des Fehlerzustands kann das DRFS aber so ausgelegt werden, dass die Zugehörigkeit bei einem fortdauernden leichten Ausschlag der Residuen immer weiter zunimmt. Die leichten Einbrüche in den Zugehörigkeiten sind mit Transienten in den Residuen aufgrund von Führungsgrößenänderungen zu erklären. Trotz des nur geringen Ausschlags des Residuums und der Transienten wird der Fehler aber erkannt. Blockiert zusätzlich der Ventilschieber, so treten aufgrund der damit verbundenen Nichtlinearität und der Reglerauslegung

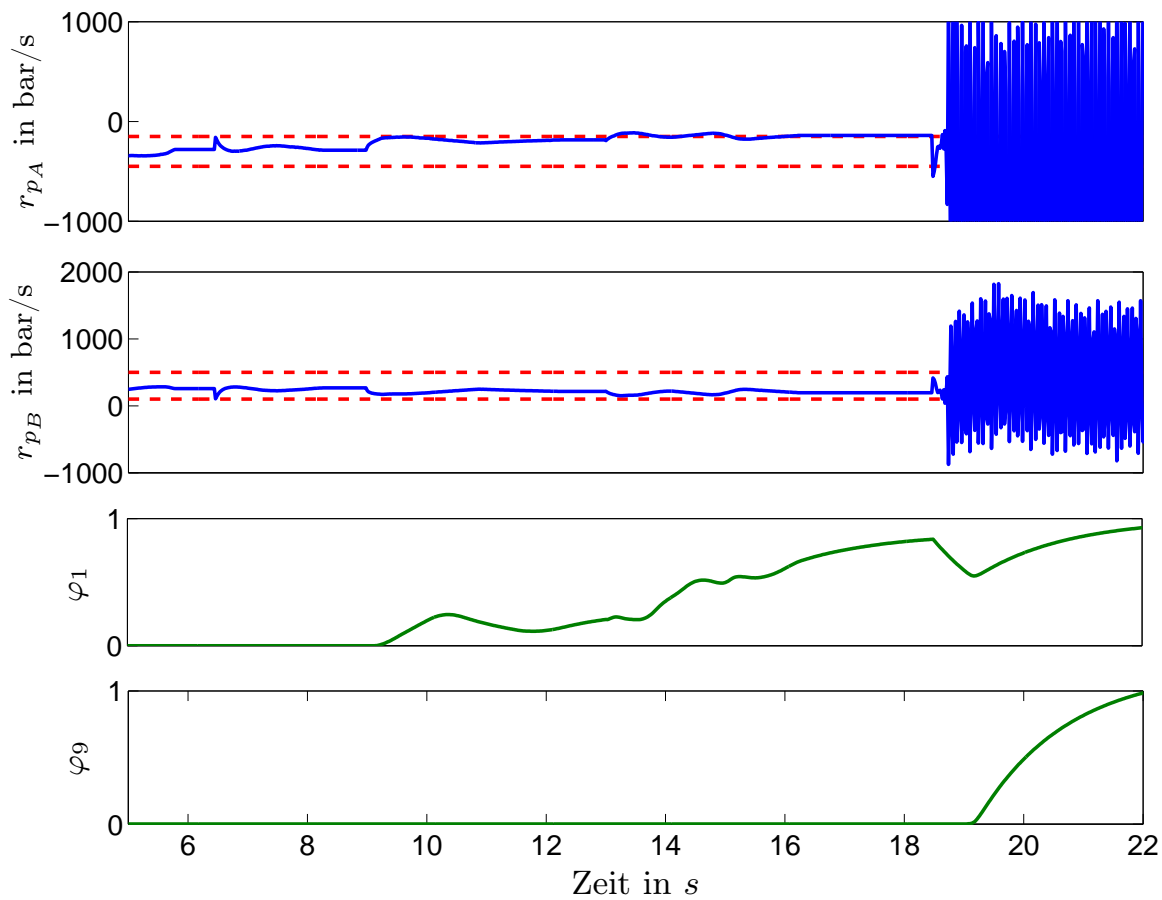


Abbildung 8.14: Residuenverläufe r_{p_A} und r_{p_B} (durchgezogen) mit Kernpositionen (gestrichelt) sowie Ergebnisse der Fehlerisolation (unten) bei Fehlerszenario 3. Beide Fehler φ_1 und φ_9 werden korrekt isoliert.

Grenzzyklen in beiden Residuen r_{p_A} und r_{p_B} auf. Hier zeigt das Verhalten der realen Strecke vom Simulationsmodell abweichendes Verhalten, da nun Grenzzyklen auftreten (vgl. Tabelle 8.4). Diese Abweichung lässt sich mit einer veränderten Last am Zylinder erklären. Während im Modell von einem Feder-Masse-Dämpfer-System als Last ausgegangen wird, war die Last während der Messungen abgekoppelt. Die verringerte Last macht das Gesamtsystem somit empfindlicher für Grenzzyklen. Nach entsprechender Anpassung der Regelbasis zeigt das DRFS beide aufgetretenen Fehler φ_1 und φ_9 korrekt an.

Szenario 4: Doppelfehler mit Leckage und anschließendem Ausfall des Kolbenpositionssensors (Messung)

Abschließend wird als weiterer Doppelfehler das Auftreten einer inneren Leckage am Kolben zum Zeitpunkt $t = 10,2\text{s}$ und ein anschließend zum

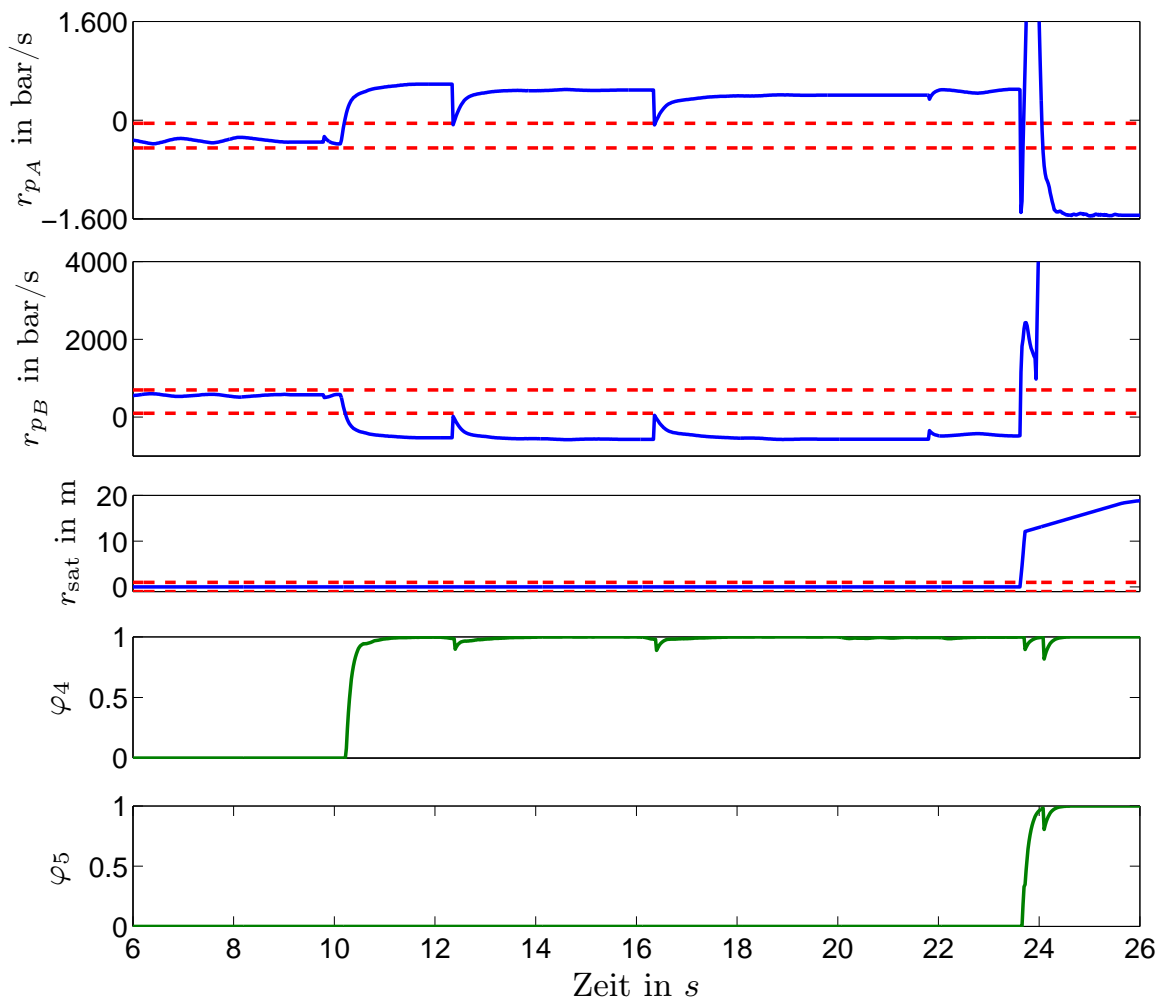


Abbildung 8.15: Residuenverläufe r_{p_A} , r_{p_B} und r_{sat} (durchgezogen) mit Kernpositionen (gestrichelt) sowie Ergebnisse der Fehlerisolation (unten). Beide Fehler φ_4 und φ_5 werden nach dem Auftreten korrekt isoliert.

Zeitpunkt $t = 23,7\text{s}$ auftretender Ausfall des Positionssensors am Zylinderkolben untersucht. Die Verläufe der Residuen r_{p_A} , r_{p_B} sowie der Fehler φ_4 und φ_5 sind in Abbildung 8.15 zu sehen. Bei Auftreten der Leckage schlägt r_{p_A} positiv und r_{p_B} negativ aus. Der Fehler φ_5 wird korrekt angezeigt. Die leichten Einbrüche in den Zugehörigkeiten sind wieder durch die Transienten in den Residuen, die dann leicht über die Kernpositionen hinausgehen, bedingt. Tritt zusätzlich Fehler φ_4 auf, so wechseln nach einer kurzen transienten Phase die Residuen das Vorzeichen. Außerdem schlägt r_{sat} aus. Das DRFS erkennt nun beide Fehler. Der kurze Einbruch bei Fehler φ_5 ist auf die Transienten beim Eintreten von Fehler φ_4 zurückzuführen.

8.2.4 Fazit

Dieser Abschnitt beschäftigte sich mit der auf rekurrenten Fuzzy-Systemen basierenden Fehlerdiagnose an einem Ventilsystem. Die Ergebnisse zeigen, dass auch mit einer vergleichsweise kleinen Zahl von Residuen eine leistungsfähige Diagnose möglich ist. Insbesondere die Möglichkeit, Fehler mit dynamischen Residuenverläufe zu isolieren, macht DRFS zu einer sinnvollen Erweiterung von SFS. Die Ergebnisse am realen System unterstreichen die praktische Einsetzbarkeit der Methode.

Bisher wurde die Fehlerdiagnose an nur einem Ventil durchgeführt. Das System lässt sich aber sehr einfach um das zweite Ventil des Duplex-Systems erweitern. Dies ist vor allem zur fehlertoleranten Auslegung des Systems notwendig. Fällt ein Ventil aus, so kann das System zumindest eine bestimmte Zeit weiterbetrieben werden. In diesem Fall können allerdings die im abgeschalteten Ventil berechneten Residuen nicht mehr zur Diagnose verwendet werden. Da das RFS den Fehlerzustand zurückführt, kann die Regelbasis dann automatisch auf den noch verlässlichen Teil der Residuen umschalten. Auch für solche Anwendungen kann also ein DRFS eingesetzt werden.

9 Zusammenfassung

Ziel dieser Arbeit war es, die Anwendung von rekurrenten Fuzzy-Systemen zur Modellierung dynamischer Systeme und zur Fehlerdiagnose zu untersuchen. Die Verwendung von rekurrenten Fuzzy-Systemen bietet hierbei insbesondere den Vorteil, dass die erhaltenen Systeme in einer Regelbasis darstellbar und somit linguistisch interpretierbar sind. Dies erhöht die Transparenz der Modelle im Vergleich zu reinen Black-Box Identifikationsverfahren. Gleichzeitig erlauben RFS über die datengestützte Modellbildung hinaus auch die Integration verschiedener Formen von Expertenwissen. Letzteres wird bei Fuzzy-Systemen vor allem durch Regeln beschrieben. Durch die in Kapitel 3 eingeführte alternative Darstellung RFS als hybrides polynomiales System wird die Integration weiterer Formen von Vorwissen, wie Systemeigenschaften in bestimmten Arbeitsbereichen und die Relevanz von möglichen Eingangs- und Zustandsgrößen ermöglicht. In Kapitel 4 wird die hybride Systemdarstellung außerdem zur Analyse der Stabilität rekurrenter Fuzzy-Systeme verwendet und ermöglicht die Betrachtung zeitkontinuierlicher und zeitdiskreter RFS in einem gemeinsamen Rahmen.

Der systematische Entwurf von zeitkontinuierlichen Fuzzy-Systemen zur Modellierung dynamischer Systeme ist Gegenstand von Kapitel 5. Die vorgestellten Verfahren kombinieren hierbei verschiedene Formen von Vorwissen, wie regelbasiertes Expertenwissen, Vorwissen über Systemeigenschaften in bestimmten Arbeitsbereichen und relevante Eingangs- und Zustandsgrößen, sowie Messdaten des betrachteten Systems. Sie ermöglichen sowohl die alleinige Optimierung der Parameter als auch, sofern nur sehr geringes Vorwissen vorhanden ist, die Optimierung der Struktur des RFS. Durch verschiedene Maßnahmen, wie die Reduktion der Kernpositionsableitungen und Reduktion der Regeln, kann die Verständlichkeit des Modells und die Generalisierungsfähigkeit deutlich erhöht werden. Im Ergebnis erhält der Anwender ein vollständiges RFS, das die Dynamik des zu modellierenden Prozess linguistisch beschreiben und in einer Regelbasis darstellen kann. Diese transparente Darstellung ermöglicht somit die Plausibilisierung des erhaltenen Modell durch den Anwender.

Neben zeitkontinuierlichen rekurrente Fuzzy-Systemen können auch zeitdiskrete rekurrenten Fuzzy-System zur Modellierung verwendet werden, wie in Kapitel 6 gezeigt wird. Allerdings eignen sie sich aufgrund ihres speziellen strukturellen Aufbaus insbesondere zur qualitativen Modellierung dynamischer Systeme. Die Möglichkeit, die Funktionsweise DRFS durch einen linguistischen Automaten näherungsweise wiederzugeben, erlaubt eine Systembetrachtung auf einer abstrakteren Darstellungsebene. Die datengestützte Modellbildung erfolgt allerdings mit einer anderen Methodik als bei KRFS. So lässt sich das Modellierungsproblem als gemischt-ganzzahliges Optimierungsproblem darstellen, zu dessen Lösung ein zweischrittiges Verfahren vorgeschlagen wird. Zunächst wird die Regelbasis des System mittels ganzzahliger Optimierung ermittelt. Anschließend werden die freien Parameter durch ein Gradientenverfahren optimiert. Diese Vorgehensweise ermöglicht neben der Anwendung zur Modellbildung auch die datengestützte Optimierung von sequentiellen Mustererkennern und Systemen zur Fehlerdiagnose.

Die Anwendung von RFS zur Fehlerdiagnose wird in Kapitel 7 diskutiert. Da moderne Verfahren zur Fehlerdiagnose zunehmend Prozessmodelle verwenden, ergibt sich als Anwendungsfeld direkt die modellbasierte Residuengenerierung mithilfe der in den vorigen Kapiteln entwickelten Modellierungsverfahren. Des Weiteren können zeitdiskrete rekurrente Fuzzy-Systeme aber auch zur Fehlerisolation eingesetzt werden. Sie stellen somit eine Erweiterung der bisher üblichen statischen Fuzzy-Systeme dar und erlauben zusätzlich die Fehlerisolation bei dynamischen Residuenverläufen. Solche liegen dann vor, wenn zur Entscheidung, welcher Fehler aufgetreten ist, nicht nur der resultierende Ausschlag sondern auch die zeitliche Entwicklung der Residuen berücksichtigt werden muss. Außerdem werden die Möglichkeiten, mehrere nacheinander auftretende Fehler zu isolieren, durch die Verwendung von DRFS deutlich verbessert.

Um die Anwendung der Verfahren zur Modellierung und Fehlerdiagnose auch in der Praxis zu zeigen, werden in Kapitel 8 zwei Beispiele behandelt. Zunächst wird die Modellierung einer Destillationskolonne mithilfe zeitdiskreter rekurrenter Fuzzy-Systeme dargestellt. Das Modell kann das dynamische Verhalten sowohl qualitativ als auch quantitativ sehr gut nachbilden. Anschließend wird die Fehlerisolation mithilfe zeitdiskreter rekurrenter Fuzzy-Systeme als Teil der Fehlerdiagnose an einer hydraulischen Servoachse mit Duplex-Ventilsystem vorgestellt. Der Ansatz weist eine deutliche Verbesserung der Fehlerisolation bei Fehlern mit dynamischen Residuenverläufen sowie bei mehreren im System auftretenden Fehlern auf.

Nachdem sich diese Arbeit auf die Modellierung dynamischer Systeme mit rekurrenten Fuzzy-Systemen konzentrierte, kann in weiterführenden Arbeiten die Regelung mit RFS untersucht werden. Hierbei kann zum einen die Regelung von dynamischen Systemen, die durch rekurrente Fuzzy-Systeme beschrieben sind, im Mittelpunkt stehen. Zum anderen kann ein rekurrentes Fuzzy-System auch als Regler eingesetzt werden. Basierend auf den Stabilitätsuntersuchungen in Kapitel 4 liegt ein möglicher Ansatzpunkte in der Verwendung von polynomialen Regelungen. Ein weiterer Ansatz, der insbesondere für DRFS interessant ist, kann die modellprädiktive Regelung liefern.

Auch im Bereich Modellierung sind weitergehende Untersuchungen denkbar. Um die Abhängigkeit von gemessenen Zustandsgrößen aufzulösen, kann der bestehende Ansatz zur Parameteroptimierung so erweitert werden, dass nur noch Ein- und Ausgangsgrößen benötigt werden. Außerdem ist es wünschenswert, die Parameteroptimierung durch Ausnutzung der polynomialen Darstellung weiter zu konvexifizieren. Dann können SOS-Methoden auch zur Parameteroptimierung angewendet werden. Aufgrund der großen Anzahl an Parametern ist dies allerdings mit den heutigen Solvern nur für sehr kleine Systeme möglich. Die Untersuchung weiterer Anwendungsfelder der Modellierung mit RFS, beispielsweise ein Einsatz im Bereich der Bildverarbeitung oder bei Wirtschaftsprognosen, ist ebenfalls interessant.

Bei der Modellierung von DRFS ist es außerdem möglich durch Relaxierung der Rückkopplungskorrespondenz die bisher sequentielle Identifikation von Regelbasis und Parametern in ein gemeinsames Optimierungsproblem zu überführen. So können weiterhin sehr gut interpretierbare Modelle identifiziert werden, die allerdings aufgrund der etwas freieren Wahl der Parameter eine bessere Approximationsgüte ermöglichen.

A Berechnungen

A.1 Singulärwertzerlegung zur Lösung von Least-Squares Problemen

Die Singulärwertzerlegung bezeichnet die Zerlegung einer Matrix in ein Produkt aus drei Matrizen mit speziellem Aufbau [63, 201]. Sie kann u.a. zur Inversion von schlecht konditionierten Matrizen, wie sie z.B. bei Least-Squares Problemen gemäß Abschnitt 5.2.1 auftreten, verwendet werden. Es wird also die Lösung des Optimierungsproblems

$$\min_{\boldsymbol{\theta}} (\mathbf{z} - \boldsymbol{\Psi}\boldsymbol{\theta})^T (\mathbf{z} - \boldsymbol{\Psi}\boldsymbol{\theta}), \quad (\text{A.1})$$

mit $\boldsymbol{\Psi} \in \mathbb{R}^{N \times M}$ und $\boldsymbol{\theta} \in \mathbb{R}^{M \times 1}$ betrachtet. Hierbei ist M die Zahl der zu optimierenden Parameter und N die Zahl der Datensätze. Die Lösung $\boldsymbol{\theta}^*$ von (A.1) erfüllt die Gleichung

$$\boldsymbol{\Psi}^T \boldsymbol{\Psi} \boldsymbol{\theta}^* = \boldsymbol{\Psi}^T \mathbf{z}. \quad (\text{A.2})$$

Um diese nach den gesuchten Parametern $\boldsymbol{\theta}^*$ aufzulösen, wird die Inversion der Matrix $\mathbf{W} = \boldsymbol{\Psi}^T \boldsymbol{\Psi}$ benötigt. Diese wird oftmals durch die schlechte Konditionierung der Matrix \mathbf{W} erschwert, d.h., eine Inversion ist aufgrund numerischer Probleme nicht einfach möglich. Deshalb teilt man die Matrix mithilfe der Singulärwertzerlegung in drei Matrizen gemäß

$$\mathbf{W} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T$$

auf, wobei $\mathbf{U} \in \mathbb{R}^{N \times N}$ und $\mathbf{V} \in \mathbb{R}^{M \times M}$ orthogonale Matrizen sind und $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_M) \in \mathbb{R}^{N \times M}$ eine Diagonalmatrix ist, deren Diagonalelemente $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_M$ die Singulärwerte von \mathbf{W} sind. Bezeichnet man nun die Spalten von \mathbf{U} und \mathbf{V} mit \mathbf{u}_i und \mathbf{v}_i , so ergibt sich als Lösung von (A.2)

$$\boldsymbol{\theta}^* = \sum_{i=1}^r \frac{\mathbf{u}_i^T \mathbf{z}}{\sigma_i} \mathbf{v}_i,$$

wobei r der Rang der Matrix \mathbf{W} ist. Es ergibt sich also mithilfe der Singulärwertzerlegung ein einfach zu berechnender Ausdruck für $\boldsymbol{\theta}^*$. Allerdings kann bei schlecht konditionierten Matrizen, wenn also $\sigma_1 \geq \dots \sigma_r \geq \sigma_{r+1} \approx \dots \approx \sigma_M \approx 0$ gilt, der Rang nurmehr näherungsweise bestimmt werden. Hierzu wählt man einen Toleranzwert δ_{tol} so, dass $\sigma_1 \geq \dots \sigma_{\hat{r}} \geq \delta_{tol} \geq \sigma_{\hat{r}+1} \geq \dots \geq \sigma_M$ gilt. Dann berechnet sich $\boldsymbol{\theta}^*$ gemäß

$$\boldsymbol{\theta}^* = \sum_{i=1}^{\hat{r}} \frac{\mathbf{u}_i^T \mathbf{z}}{\sigma_i} \mathbf{v}_i.$$

Für die Least-Squares Optimierung bedeutet dies anschaulich, dass nur noch die Informationen aus den Datensätzen, die nicht redundant sind (ausgedrückt durch einen großen Singulärwert), zur Berechnung der Lösung herangezogen werden. Mit der Wahl von δ_{tol} kann eingestellt werden, wieviel Information als redundant eingestuft wird.

A.2 Matrizendarstellung eines Hyperquaders

Die mathematische Darstellung eines Hyperquaders kann alternativ auch über Matrizen bestimmt werden. Da die Matrix $\mathbf{\Gamma}$ in Gl. (3.16) mithilfe dieser Darstellung bestimmt wird, soll sie hier angegeben werden. Für den l -ten Hyperquader gilt in der Matrixdarstellung

$$\delta \mathbf{x} = \mathbf{S}_{j_l} \boldsymbol{\Delta}^{(n+m)} \boldsymbol{\Omega}^{(n+m)} \boldsymbol{\omega}^{(n+m)}. \quad (\text{A.3})$$

Der Vektor $\boldsymbol{\omega}$ und die Matrizen \mathbf{S}_{j_l} , $\boldsymbol{\Delta}$, $\boldsymbol{\Omega}$ werden rekursiv für $k = 0, \dots, n+m-1$ durch folgende Konstruktionsvorschrift gebildet:

$$\begin{aligned} \mathbf{S}_{j_l} &= \mathbf{S}_{\mathbf{w}(\mathbf{V}^{(n+m)})}^{\delta \mathbf{x}}, \\ \mathbf{V}^{(k+1)} &= \begin{bmatrix} \mathbf{V}^{(k)} & \mathbf{V}^{(k)} + \mathbf{E}_{k+1} \end{bmatrix}, \quad \mathbf{V}^{(0)} = \mathbf{j}_l, \\ \mathbf{S}_{\mathbf{w}(\mathbf{A})}^{\delta \mathbf{x}} &= \begin{bmatrix} \mathbf{S}_{\mathbf{w}(\mathbf{a}_1)}^{\delta \mathbf{x}} & \dots & \mathbf{S}_{\mathbf{w}(\mathbf{a}_p)}^{\delta \mathbf{x}} \end{bmatrix}, \quad \mathbf{A} \in \mathbb{R}^{n \times p}, \quad p = 2^n, \\ \boldsymbol{\Delta}^{(k+1)} &= \begin{bmatrix} \boldsymbol{\Delta}^{(k)} & -\boldsymbol{\Delta}^{(k)} \\ \mathbf{0} & \boldsymbol{\Delta}^{(k)} \end{bmatrix}, \quad \boldsymbol{\Delta}^{(0)} = \mathbf{1}, \\ \boldsymbol{\Omega}^{(k+1)} &= \begin{bmatrix} \boldsymbol{\Omega}^{(k)} & \mathbf{0} \\ \frac{-s_{v_{k+1},l}^{z_{k+1}}}{s_{v_{k+1},l+1}^{z_{k+1}} - s_{v_{k+1},l}^{z_{k+1}}} \boldsymbol{\Omega}^{(k)} & \frac{1}{s_{v_{k+1},l+1}^{z_{k+1}} - s_{v_{k+1},l}^{z_{k+1}}} \boldsymbol{\Omega}^{(k)} \end{bmatrix}, \quad \boldsymbol{\Omega}^{(0)} = \mathbf{1}, \\ \boldsymbol{\omega}^{(k+1)} &= \begin{bmatrix} \boldsymbol{\omega}^{(k)} \\ \boldsymbol{\omega}^{(k)} z_{k+1} \end{bmatrix}, \quad \boldsymbol{\omega}^{(0)} = \mathbf{1}, \end{aligned}$$

wobei \mathbf{a}_i den i -ten Spaltenvektor von \mathbf{A} bezeichnet und

$$\mathbf{E}_k = \begin{bmatrix} \delta_{k1} & \cdots & \delta_{k1} \\ \delta_{k2} & \cdots & \delta_{k2} \\ \vdots & \ddots & \vdots \\ \delta_{kn} & \cdots & \delta_{kn} \end{bmatrix} \in \mathbb{R}^{n \times 2^{k-1}},$$

mit dem Kronecker delta δ_{ij} . Hieraus folgt schließlich

$$\mathbf{\Gamma}(\mathbf{s}_j^{\mathbf{x}}, \mathbf{s}_q^{\mathbf{u}}) = \mathbf{\Delta}^{(n+m)} \mathbf{\Omega}^{(n+m)}.$$

Der Nachweis, dass obige Darstellung (A.3) mit der linguistischen Darstellung (3.11) äquivalent ist, wird in [173] mithilfe vollständiger Induktion geführt.

B Modellparameter

B.1 Parameter und Einstellwerte der Fuzzy-Modelle

Dieser Anhang enthält die Parameter, d.h. die Kernpositionen und Kernpositionsableitungen, der Simulationsbeispiele Bioreaktor (Abschnitt 5.5.1), Rührkesselreaktor (Abschnitt 5.5.2) und thermofluidischer Prozess (Abschnitt 6.4) sowie der Destillationskolonne (Abschnitt 8.1). In den Algorithmen zur Modellierung des Bioreaktors mit KRFS wurden die Einstellparameter $\delta_{tol} = 10^{-11}$, $\alpha_{gr} = 0,1$, $\gamma_0 = 9$, $\alpha_{ASK}^* = 0,3$, $\beta^* = 0,5$, $R_w = 20$ verwendet.

Tabelle B.1: Kernpositionen und Kernpositionsableitungen des KRFS zur Modellierung des Bioreaktors.

	$s_1^{x_1}$	$s_2^{x_1}$	$s_1^{x_2}$	$s_2^{x_2}$	$s_3^{x_2}$	$s_4^{x_2}$	$s_5^{x_2}$	$s_6^{x_2}$	s_1^u	s_2^u
	0,08	0,27	0,44	0,72	0,88	0,95	0,98	0,99	0,32	1,11
i	$s_1^{\dot{x}_i}$	$s_2^{\dot{x}_i}$	$s_3^{\dot{x}_i}$	$s_4^{\dot{x}_i}$	$s_5^{\dot{x}_i}$	$s_6^{\dot{x}_i}$	$s_7^{\dot{x}_i}$	$s_8^{\dot{x}_i}$	$s_9^{\dot{x}_i}$	$s_{10}^{\dot{x}_i}$
1	-0,21	-0,20	-0,18	-0,11	-0,05	0,01	0,06	0,08	0,15	0,34
2	-0,90	-0,87	-0,54	-0,33	-0,07	0,11	0,47	0,90	1,13	1,18

Tabelle B.2: Parameter des DRFS zur Modellierung des thermofluidischen Prozesses.

$s_1^{x_1}$	$s_2^{x_1}$	$s_3^{x_1}$	$s_1^{x_2}$	$s_2^{x_2}$	$s_3^{x_2}$	$s_1^{u_1}$	$s_2^{u_1}$	$s_1^{u_2}$	$s_2^{u_2}$
0,23	0,36	0,49	272,83	293,01	315,09	0	0,95	0	6,15

Tabelle B.3: Parameter des thermofluidischen Prozesses.

Querschnittsfläche Tank A	0,07 m ²
Ausflussparameter K_A	$1,59 \cdot 10^{-5}$ m ³ /m
Systemkoeffizient k_h	0,84 1/(Wm ³)
Gesamtheizleistung P_{el}	3000 W
Wärmekapazität von Wasser c_p	4180 J/(kgK)
Dichte von Wasser (20°C) ρ	998 kg/m ³
Umgebungstemperatur ϑ_U	293.15 K

Die Einstellparameter der Modellierung des Rührkesselreaktors sind $\delta_{tol} = 10^{-11}$, $\alpha_{gr} = 0,1$, $\beta^* = 0,5$, $R_w = 120$ $\alpha_{EA} = 0$.

Tabelle B.4: Kernpositionen des KRFS zur Modellierung des Rührkesselreaktors. Auf die Angabe der Kernpositionsableitungen wird aufgrund der großen Zahl (160 Kernpositionsableitungen pro Dimension) verzichtet.

$s_1^{x_1}$	$s_2^{x_1}$	$s_1^{x_2}$	$s_2^{x_2}$	$s_1^{x_3}$	$s_2^{x_3}$	$s_3^{x_3}$	$s_4^{x_3}$	$s_5^{x_3}$
0,59	2,45	0,51	1,27	98,87	118,82	126,55	129,66	135,62
$s_1^{x_4}$	$s_2^{x_4}$	$s_1^{u_1}$	$s_2^{u_1}$	$s_1^{u_2}$	$s_2^{u_2}$			
92,14	137,01	3,44	39,09	-8883	1231			

Tabelle B.5: Parameter des DRFS zur Modellierung der Destillationskolonne.

$s_1^{T_{12}}$	$s_2^{T_{12}}$	$s_3^{T_{12}}$	$s_1^{T_2}$	$s_2^{T_2}$	$s_3^{T_2}$	$s_1^{T_3}$	$s_2^{T_3}$	$s_3^{T_3}$	$s_1^{\Delta V_1}$
-3,58	-2,96	-1,26	21,06	22,67	23,33	24,49	24,98	26,39	-6,1
$s_1^{P_D}$	$s_2^{P_D}$	$s_3^{P_D}$	$s_1^{F_2}$	$s_2^{F_2}$	$s_3^{F_2}$	$s_1^{V_1}$	$s_2^{V_1}$	$s_3^{V_1}$	$s_2^{\Delta V_1}$
0,41	0,6	0,81	-0,046	0	0,041	66,81	73,98	90,71	8,36

Tabelle B.6: Parameter des Hydraulikmodells.

Kolbenfläche A_A	0,0013 m ²
Kolbenfläche A_B	$6,41 \cdot 10^{-4}$ m ²
Totvolumen V_{0A}	0 m ³
Totvolumen V_{0B}	$1,92 \cdot 10^{-4}$ m ³
Elastizitätsmodul E	$1,2 \cdot 10^9$ N/m ²
Lastmasse m	53 kg
Federsteifigkeit c	100,27 N/mm
Federvorspannung F_v	270 N

B.2 Symptomtabelle für die Doppelfehler

Tabelle B.7: Fehlersymptomtabelle für die Doppelfehlerfälle. Es werden die folgenden Abkürzungen verwendet: t $\hat{=}$ Transienten, gz $\hat{=}$ Grenzyklus, * $\hat{=}$ Residuen ist abhängig von beiden Fehlergrößen.

	φ_{12}	φ_{13}	φ_{14}	φ_{15}	φ_{16}	φ_{17}	φ_{18}	φ_{19}	φ_{21}	φ_{23}
r_{pA}	+	*	-	*/+	+	-	+	-	+	-
r_{pB}	+	-	gz	+/-	t	0	-	-/+	+	*
r_{sat}	0	0	+	0	0	+/-	0	+/-	0	0

	φ_{24}	φ_{25}	φ_{26}	φ_{27}	φ_{28}	φ_{29}	φ_{31}	φ_{32}	φ_{34}	φ_{35}
r_{pA}	-	-/+	t	0	+	-	*	-	-	-/*
r_{pB}	gz	+/*	+	-	*	-/+	-	*	gz	*/-
r_{sat}	+	0	0	+/-	0	+/-	0	0	+	0

	φ_{36}	φ_{37}	φ_{38}	φ_{39}	φ_{51}	φ_{52}	φ_{53}	φ_{54}	φ_{56}	φ_{57}
r_{pA}	-	+/-0	*	-	*/+	-/+	-/*	-	-/+	-
r_{pB}	-	0/+	-	-/+	+/-	+/*	*/-	+	+/-	+
r_{sat}	0	+/-	0	+/-	0	0	0	+	0	+/-

	φ_{58}	φ_{59}	φ_{61}	φ_{62}	φ_{63}	φ_{64}	φ_{65}	φ_{67}	φ_{68}	φ_{69}
r_{pA}	*/+	-	+	t	-	-	-/+	-/0	+	-
r_{pB}	*/-	-/+	t	+	-	-	+/-	0/+	-	-/+
r_{sat}	0	+/-	0	0	0	+	0	+/-	0	+/-

	φ_{81}	φ_{82}	φ_{83}	φ_{84}	φ_{85}	φ_{86}	φ_{87}	φ_{89}		
r_{pA}	+	+	*	-	*/+	+	-/0	-		
r_{pB}	-	*	-	gz	*/-	-	0/+	-/+		
r_{sat}	0	0	0	+	0	0	+/-	+/-		

B.3 Datensätze der Simulationsmodelle

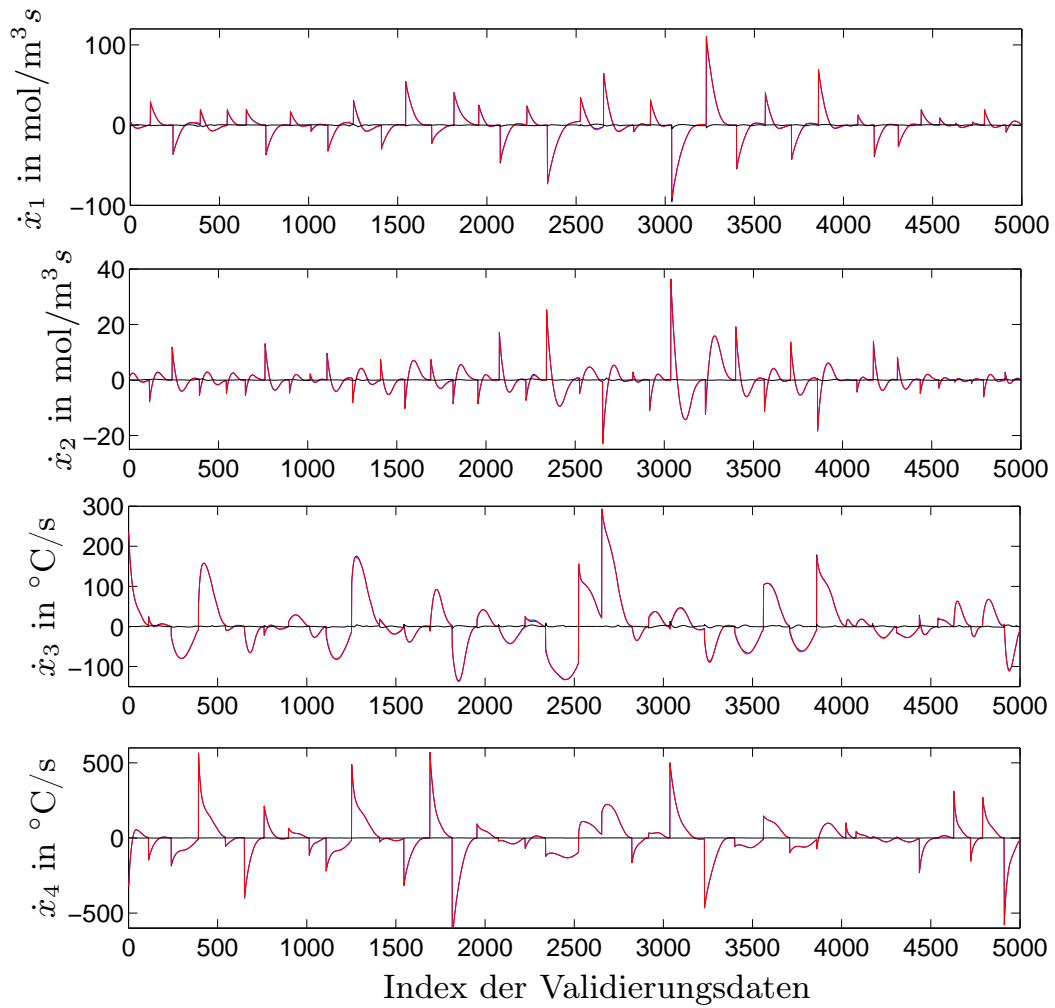


Abbildung B.1: Verläufe der Zustandsableitungen \dot{x}_1 und \dot{x}_3 für den Trainingsdatensatz bei seriell-paralleler Modellauswertung. Die Verläufe von KRFS (rot) und Beispieldaten (blau) sind nahezu identisch, wie auch der Fehler (schwarz) zeigt.

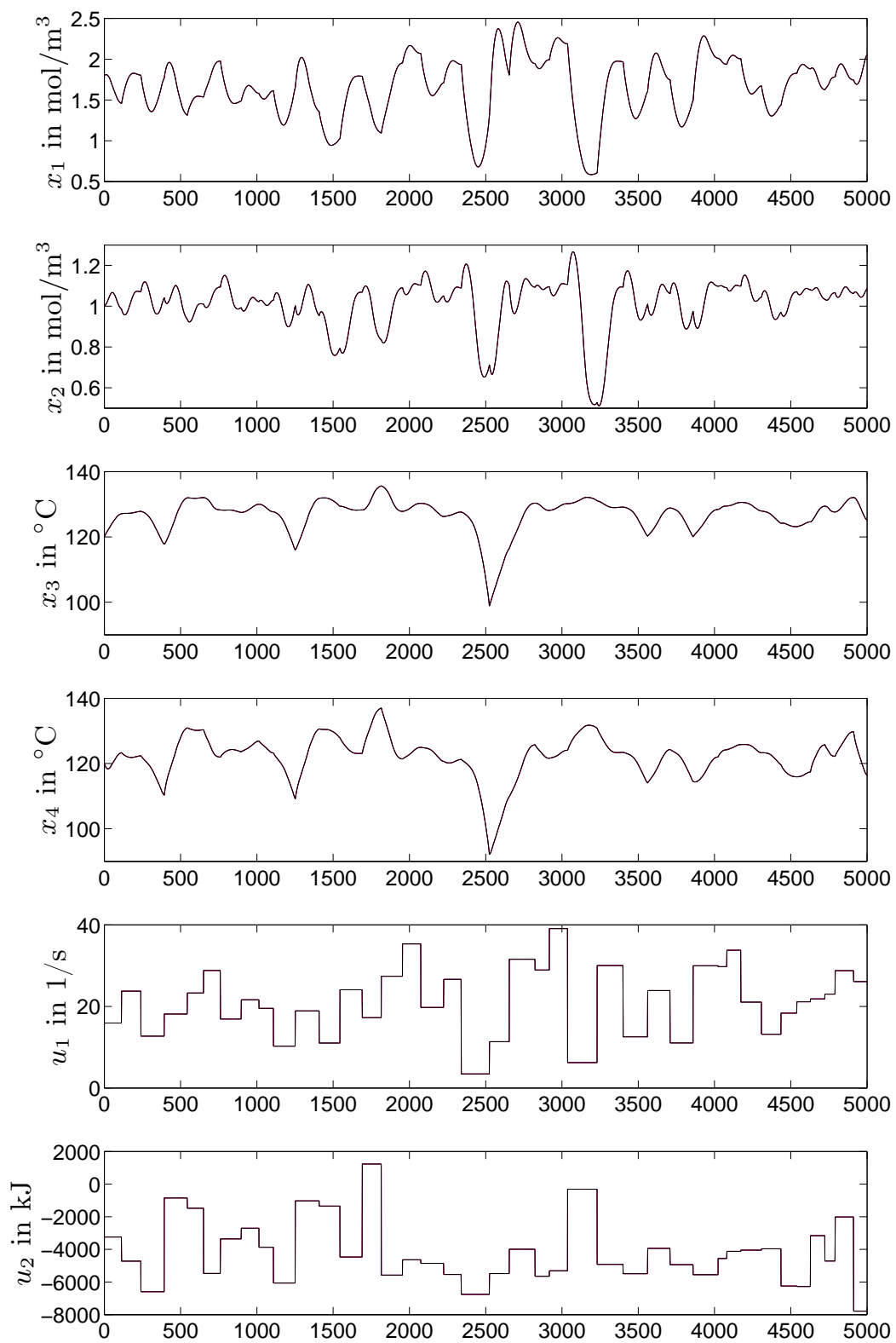
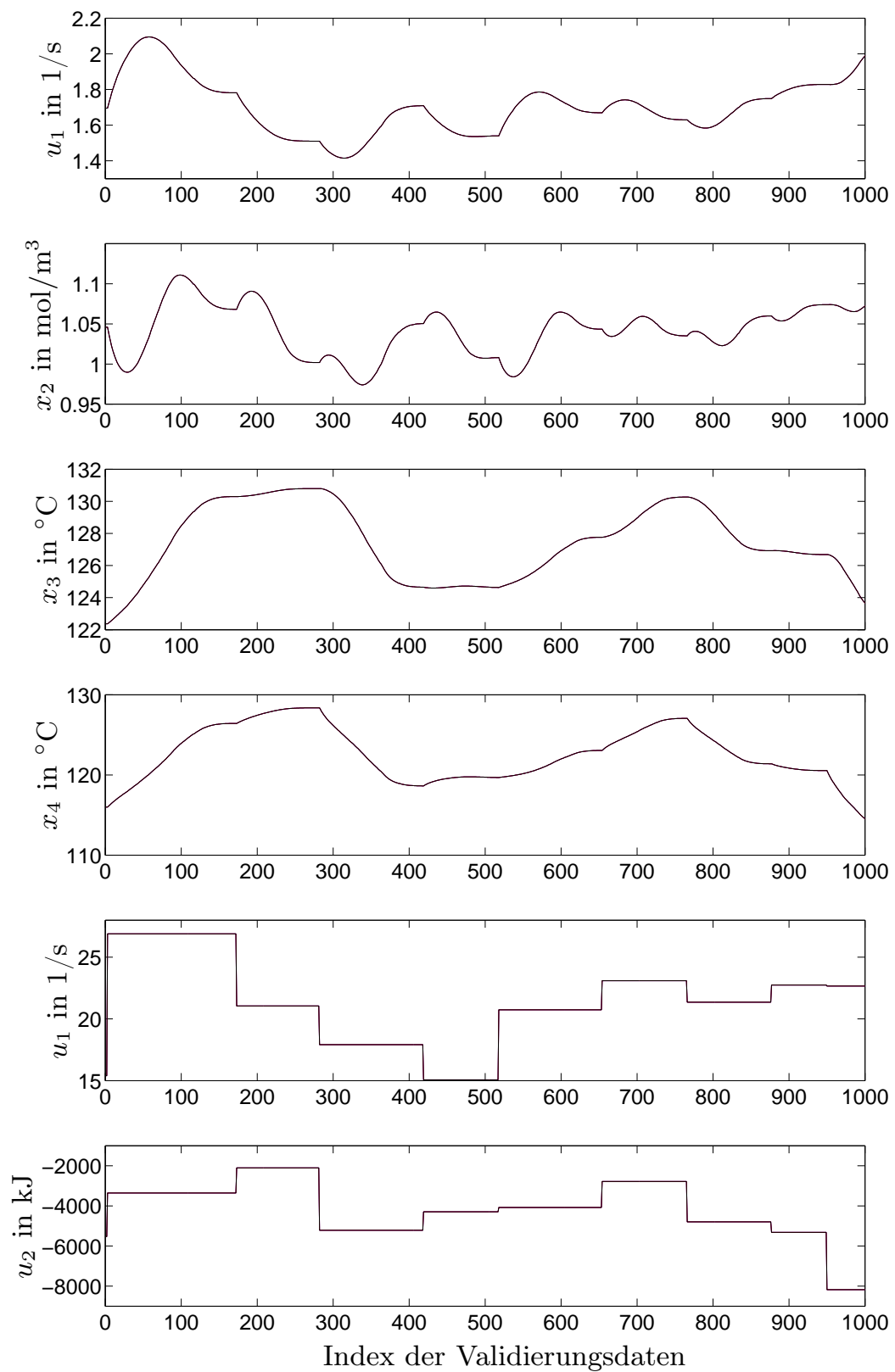


Abbildung B.2: Trainingsdatensatz für den Rührkesselreaktor.

**Abbildung B.3:** Validierungsdatensatz für den Rührkesselreaktor.

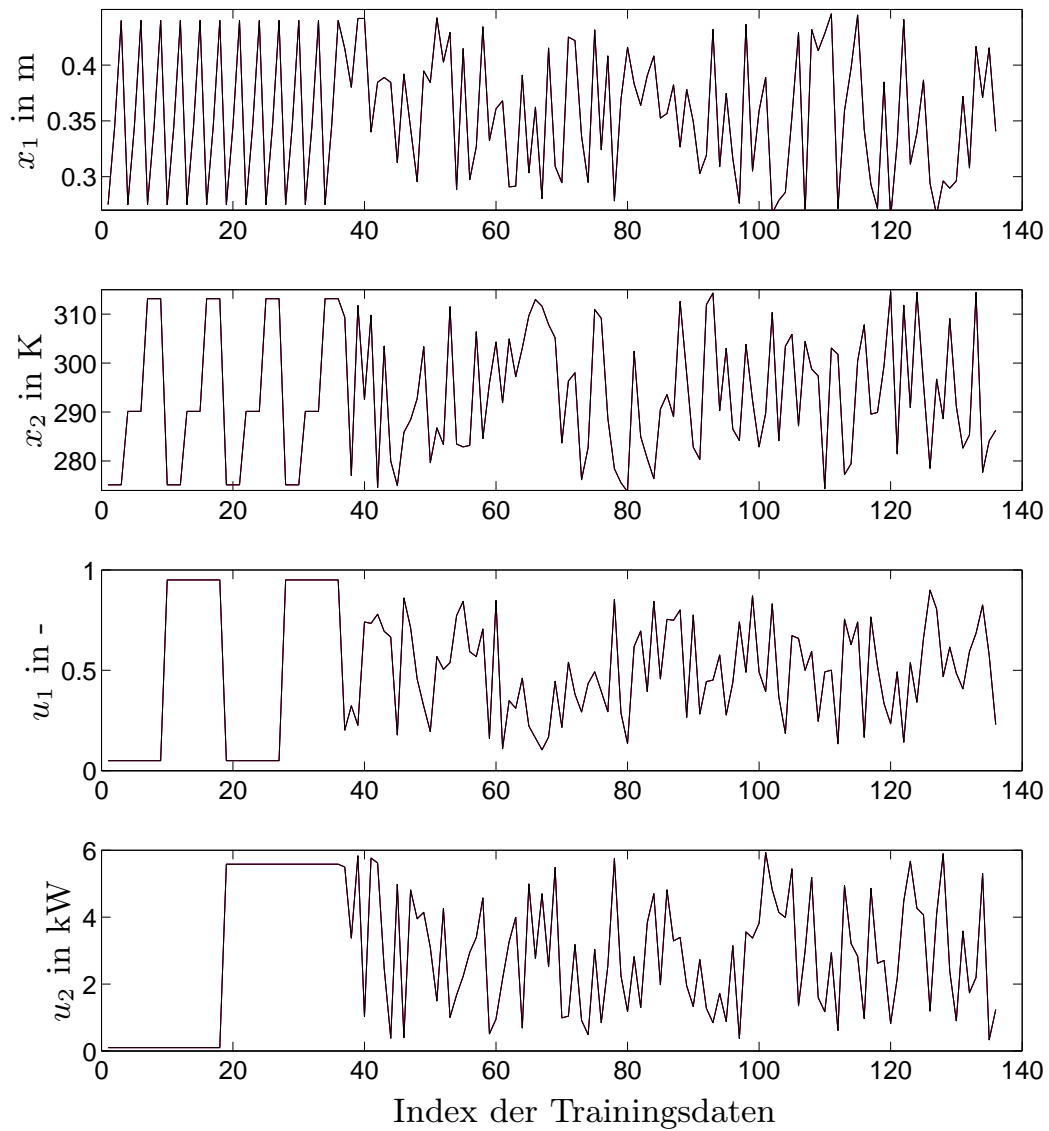


Abbildung B.4: Trainingsdatensatz für den thermofluidischen Prozess.

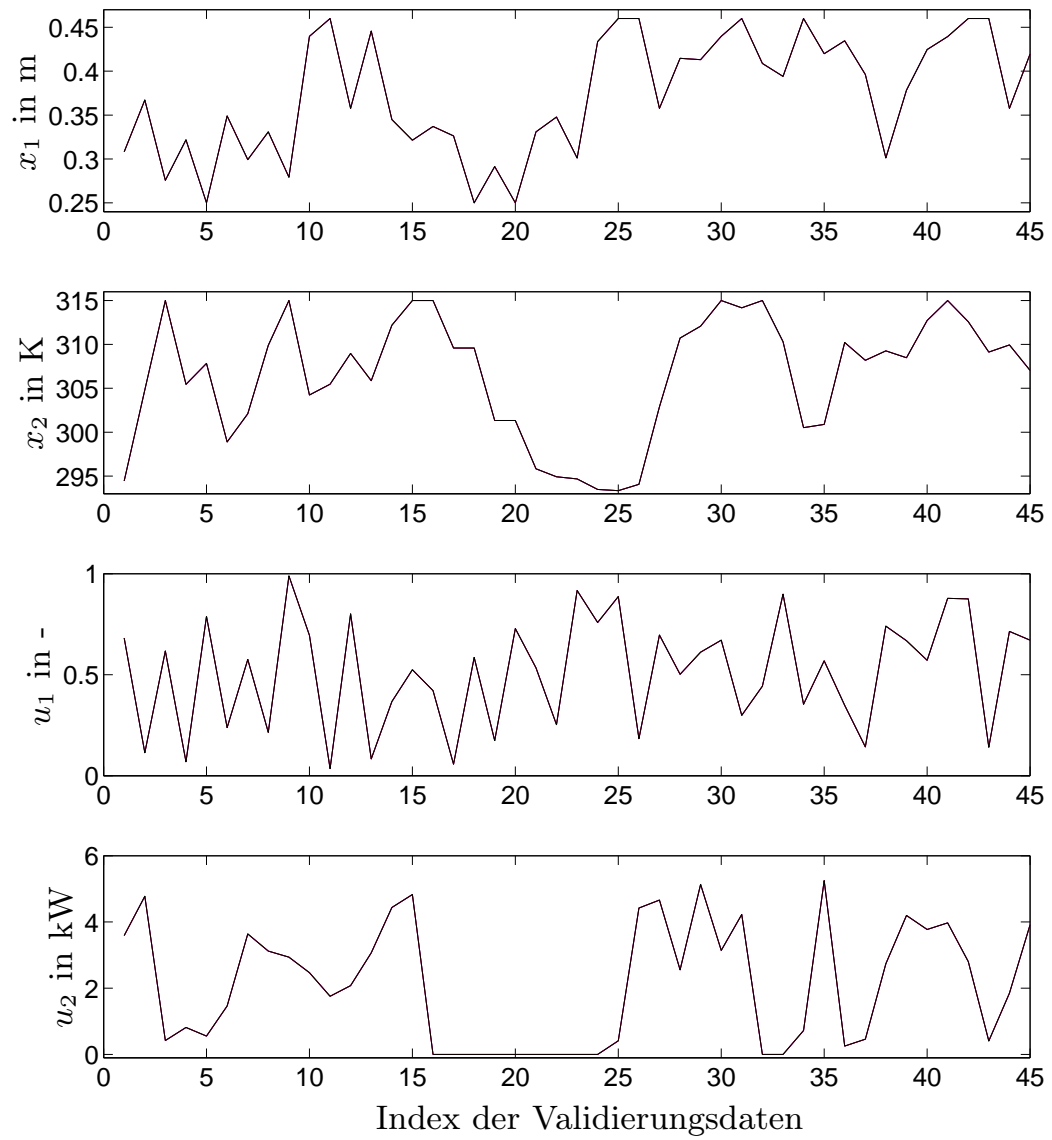


Abbildung B.5: Validierungsdatensatz für den thermofluidischen Prozess.

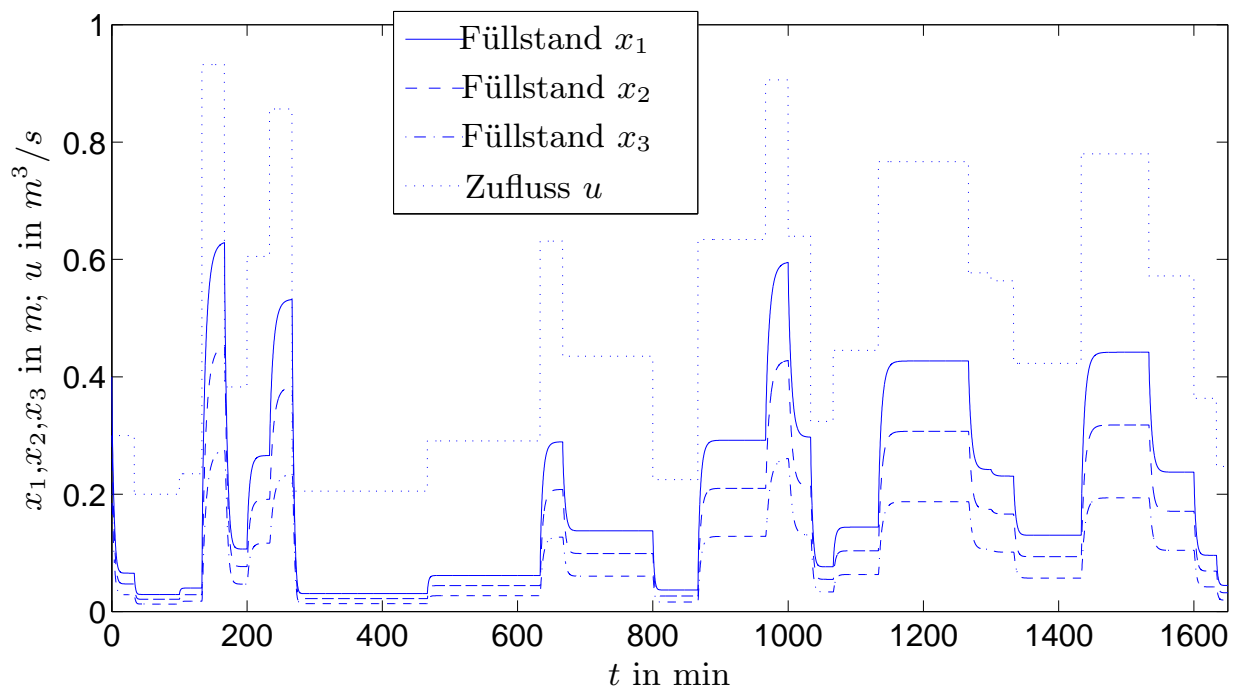


Abbildung B.6: Trainingsdatensatz für den Drei-Tank-Prozess.

Literaturverzeichnis

- [1] ABONYI, J. ; BABUŠKA, R. ; SZEIFERT, F. : Modified Gath-Geva Fuzzy Clustering for Identification of Takagi-Sugeno Fuzzy Models. In: *IEEE Trans. on System, Man, and Cybernetics - Part B: Cybernetics* 32(5) (2002), S. 612–621
- [2] ADAMY, J. : *Device for early detection of runout in continuous casting.* 1995. – EP Priority Date 03.04.1995, European Patent 0819033B1 (1998), US Patent 5,904,202 (1999)
- [3] ADAMY, J. : *Nichtlineare Regelungen.* Springer, 2009
- [4] ADAMY, J. ; FLEMMING, A. : Equilibria of Continuous-Time Recurrent Fuzzy Systems. In: *Fuzzy Sets and Systems* 157(22) (2006), S. 2913–2933
- [5] ADAMY, J. ; KEMPF, R. : Regularity and chaos in recurrent fuzzy systems. In: *Fuzzy Sets and Systems* 140(2) (2003), S. 259–284
- [6] ADAMY, J. ; SCHWUNG, A. : Modellierung dynamischer Systeme mit zeitkontinuierlichen rekurrenten Fuzzy-Systemen. In: *at-Automatisierungstechnik* 57(3) (2009), S. 129–137
- [7] ADAMY, J. ; SCHWUNG, A. : Qualitative modeling of dynamical systems employing continuous-time recurrent fuzzy systems. In: *Fuzzy Sets and Systems* 161(23) (2010), S. 3026–3043
- [8] AGRAWAL, P. ; LEE, C. ; LIM, H. ; RAMKRISHNA, D. : Theoretical investigations of dynamic behaviour of isothermal stirred tank biological reactors. In: *Chemical Engineering Science* 37(3) (1982), S. 453–462
- [9] AKAIKE, H. : A new Look at the Statistical Model Identification. In: *IEEE Trans. on Automatic Control* 19(6) (1974), S. 716–723

- [10] ALCALA, R. ; DUCANGE, P. ; HERRERA, F. ; LAZZERINI, B. ; MARCELLONI, F. : A Multi-objective Evolutionary Approach to Concurrently Learn Rule and Data Bases of Linguistic Fuzzy Rule-Based Systems. In: *IEEE Trans. on Fuzzy Systems* 17(5) (2009), S. 1106–1122
- [11] AYOUBI, M. ; ISERMANN, R. : Neuro-Fuzzy systems for diagnosis. In: *Fuzzy Sets and Systems* 89(3) (1997), S. 289–307
- [12] BABUŠKA, R. : Construction of Fuzzy Systems - Interplay between Precision and Transparency. In: *Proc. of the European Symposium on Intelligent Techniques*. Aachen, Deutschland, 2000, S. 445–452
- [13] BABUŠKA, R. ; VERBRUGGEN, H. : An Overview of Fuzzy Modeling for Control. In: *Control Engineering Practice* 4(11) (1996), S. 1593–1606
- [14] BAHAR, A. ; ÖZGEN, C. ; LEBLEBICIOGLU, K. ; HALICI, U. : Artificial Neural Network Estimator Design for the Inferential Model Predictive Control of an Industrial Distillation Column. In: *Industrial & Engineering Chemistry Research* 43(19) (2004), S. 6102–6111
- [15] BAILEY, J. ; OLLIS, D. : *Biochemical Engineering Fundamentals*. Bd. 2. McGraw-Hill, 1986
- [16] BALLE, P. ; NELLES, O. ; FÜSSEL, D. : Fault detection for non-linear process based on local linear fuzzy models in parallel and series-parallel model. In: *Proc. IFAC Symposium SAFEPROCESS*. Hull, Großbritannien, 1997, S. 1137–1142
- [17] BASTIAN, A. : Identifying fuzzy models utilizing genetic programming. In: *Fuzzy Sets and Systems* 113(3) (2000), S. 333–350
- [18] BECK, M. : *Fehlertoleranzstrategien für mechatronische Systeme*, Arbeitsgemeinschaft industrieller Forschungsvereinigungen Otto von Guericke e.V. (BMWi), Frankfurt, Abschlussbericht, 2010
- [19] BECK, M. ; MÜNCHHOF, M. ; ISERMANN, R. : Model-based fault detection and diagnosis for electromagnetic valve drives. In: *Proc. of the Dynamic System and Control Conf.* Hollywood, USA, 2009
- [20] BECK, M. ; MÜNCHHOF, M. ; SCHWUNG, A. ; ISERMANN, R. : Fehlertolerante elektrohydraulische Servoachse mit Duplex-Ventilsystem. In: *Proc. of Automation 2010*. Baden-Baden, 2010, S. 121–126

- [21] BECK, M. ; SCHWUNG, A. ; MÜNCHHOF, M. ; ISERMANN, R. : Active fault tolerant control of an electro-hydraulic servo axis with a duplex-valve-system. In: *Proc. of the 5th IFAC Symposium on Mechatronic Systems*. Cambridge, USA, 2010
- [22] BEGON, M. ; TOWNSEND, C. ; HARPER, J. : *Ecology: From Individuals to Ecosystems*. Wiley-Blackwell, 2006
- [23] BEMPORAD, A. : *Hybrid Toolbox - User's Guide*. – <http://www.dii.unisi.it/hybrid/toolbox>
- [24] BEMPORAD, A. ; MORARI, M. : Control of systems integrating logic, dynamics, and constraints. In: *Automatica* 35(3) (1999), S. 407–427
- [25] BLANKE, M. ; KINNAERT, M. ; LUNZE, J. ; STAROSWIECKI, M. : *Diagnosis and Fault-Tolerant Control*. Springer, 2006
- [26] BLEYMÜLLER, J. ; GEHLERT, G. ; GÜLICHER, H. : *Statistik für Wirtschaftswissenschaftler*. Verlag Franz Vahlen, 2004
- [27] BORTOLET, P. ; MERLET, E. ; BOVERIE, S. : Fuzzy modeling and control of an engine air inlet with exhaust gas recirculation. In: *Control Engineering Practice* 7(10) (1999), S. 1269–1277
- [28] BRANICKY, M. : Multiple Lyapunov Functions and Other Analysis Tools for Switched and Hybrid Systems. In: *IEEE Trans. on Automatic Control* 43(4) (1998), S. 475–482
- [29] CASILLAS, J. ; CORDON, O. ; HERRERA, F. ; MAGDALENA, L. : *Interpretability Issues in Fuzzy Modeling*. Springer, 2003
- [30] CHEN, J.-Q. ; XI, Y.-G. ; ZHANG, Z.-J. : A clustering algorithm for fuzzy model identification. In: *Fuzzy Sets and Systems* 98(3) (1998), S. 319–329
- [31] CORDON, O. ; GOMIDE, F. ; HERRERA, F. ; HOFFMANN, F. ; MAGDALENA, L. : Ten years of genetic fuzzy systems: current framework and new trends. In: *Fuzzy Sets and Systems* 141 (2004), S. 5–31
- [32] CORDON, O. ; HERRERA, F. : A three-stage evolutionary process for learning descriptive and approximative fuzzy logic controller knowledge bases from examples. In: *International Journal of Approximate Reasoning* 17(4) (1997), S. 369–407

- [33] CORDON, O. ; HERRERA, F. : A Two-Stage Evolutionary Process for Designing TSK Fuzzy Rule-Based Systems. In: *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics* 29(6) (1999), S. 703–715
- [34] DAIGLE, M. : *A qualitative event-based approach to fault diagnosis of hybrid systems*, Vanderbilt University, Nashville, TN, Phd Thesis, 2008
- [35] DAIGLE, M. ; KOUTSOUKOS, X. ; BISWAS, G. : A Qualitative Approach to Multiple Fault Isolation in Continuous Systems. In: *Proc. of the AAAI Conf. on Artificial Intelligence*. Vancouver, Kanada, 2007, S. 293–298
- [36] DAIGLE, M. ; KOUTSOUKOS, X. ; BISWAS, G. : A Qualitative Event-Based Approach to Continuous Systems Diagnosis. In: *IEEE Trans. on Control Systems Technology* 17(4) (2009), S. 780–793
- [37] DANIEL-BERHE, S. ; UNBEHAUEN, H. : Bilinear Continuous-Time Systems Identification via Hartley-based Modulating Functions. In: *Automatica* 34(4) (1998), S. 499–503
- [38] DE KLEER, J. ; WILLIAMS, B. : Diagnosing multiple faults. In: *Artificial Intelligence* 32(1) (1987), S. 97–130
- [39] DELGADO, M. ; GOMEZ-SKARMETA, A. F. ; MARTIN, F. : A methodology to model fuzzy systems using fuzzy clustering in a rapid-prototyping approach. In: *Fuzzy Sets and Systems* 97(3) (1998), S. 287–301
- [40] DESTERCKE, S. ; GUILLAUME, S. ; CHARNOMORDIC, B. : Building an interpretable fuzzy rule base from data using Orthogonal Least Squares - Application to a depollution problem. In: *Fuzzy Sets and Systems* 158(18) (2007), S. 2078–2094
- [41] DEXTER, A. ; BENOURETS, M. : Model-based fault diagnosis using fuzzy matching. In: *IEEE Trans. on Systems, Man, and Cybernetics - Part A* 27 (1997), S. 673–682
- [42] DIEHL, M. : *Real-Time Optimization for Large Scale Nonlinear Systems*, Universität Heidelberg, Dissertation, 2001

- [43] DIEHL, M. ; FINDEISEN, R. ; SCHWARZKOPF, S. ; USLU, I. ; ALLGÖWER, F. ; BOCK, H. ; GILLES, E. ; SCHLÖDER, J. : An Efficient Algorithm for Nonlinear Model Predictive Control of Large-Scale Systems Part 2: Experimental Evaluation. In: *at-Automatisierungstechnik* 51(1) (2003), S. 22–29
- [44] EVSUKOFF, A. ; BLANCO, A. ; GALICHET, S. : Structure identification and parameter optimization for non-linear fuzzy modeling. In: *Fuzzy Sets and Systems* 132(2) (2002), S. 173–188
- [45] FAZLE AZEEM, M. ; HANMANDLU, M. ; AHMAD, N. : Structure Identification of Generalized Adaptive Neuro-Fuzzy Inference Systems. In: *IEEE Trans. on Fuzzy Systems* 11(5) (2003), S. 666–681
- [46] FELLEISEN, M. : *Prozessleittechnik für die Verfahrensindustrie*. Oldenbourg Verlag, 2001
- [47] FINDEISEN, D. : *Ölhydraulik*. Springer, 2006
- [48] FLEMMING, A. : *Zeitkontinuierliche rekurrente Fuzzy-Systeme*, TU Darmstadt, Dissertation, 2008
- [49] FLEMMING, A. ; ADAMY, J. : Modeling solide oxide fuel cells using continuous-time recurrent fuzzy systems. In: *Engineering Applications of Artificial Intelligence* 21(8) (2008), S. 1289–1300
- [50] FLEMMING, A. ; KEMPF, R. ; ADAMY, J. : Traffic simulation employing recurrent fuzzy systems. In: *Proc. of the IASTED Intern. Conf. on Applied Simulation and Modelling*. Palma de Mallorca, Spanien, 2007, S. 497–502
- [51] FLOUDAS, C. : *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Oxford University Press, 1995
- [52] FORBUS, K. : Qualitative Process Theory / MIT Artificial Intelligence Laboratory. 1984 (789). – Forschungsbericht
- [53] FRANK, P. : Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-based Redundancy - A Survey and Some New Results. In: *Automatica* 26(3) (1990), S. 459–474
- [54] FRANK, P. ; KÖPPEN-SELIGER, B. : Fuzzy Logic and Neural Network Applications to Fault Diagnosis. In: *International Journal of Approximate Reasoning* 16(1) (1997), S. 67–88

- [55] FRANK, P. ; MARCU, T. : Fuzzy Techniques in Fault Detection, Isolation and Diagnosis. In: VERBRUGGEN, H. (Hrsg.) ; BABUŠKA, R. (Hrsg.): *Fuzzy Logic Control: Advances in Applications*. World Scientific, 1999, S. 135–159
- [56] FRANKE, D. : *Sequentielle Systeme*. Vieweg, 1994
- [57] FÜSSEL, D. ; BALLE, P. ; ISERMANN, R. : Closed loop fault diagnosis based on a nonlinear process model and automatic fuzzy rule generation. In: *Proc. of the IFAC Symposium SAFEPROCESS*. Hull, Großbritannien, 1997, S. 359–364
- [58] GACTO, M. ; ALCALA, R. ; HERRERA, F. : Integration of an Index to Preserve the Semantic Interpretability in the Multi-Objective Evolutionary Rule Selection and Tuning of Linguistic Fuzzy Systems. In: *IEEE Trans. on Fuzzy Systems* 18(3) (2010), S. 515–531
- [59] GAMA, C. ; EVSUKOFF, A. ; WEBER, P. ; EBECKEN, N. : Parameter Identification of Recurrent Fuzzy Systems with Fuzzy Finite-State Automata Representation. In: *IEEE Trans. on Fuzzy Systems* 16(1) (2008), S. 213–224
- [60] GERTLER, J. : Fault detection and isolation using parity relations. In: *Control Engineering Practice* 5(5) (1997), S. 653–661
- [61] GERTLER, J. ; SINGER, D. : A new structural framework for parity equation-based failure detection and isolation. In: *Automatica* 26(2) (1990), S. 381–388
- [62] GOEBEL, R. ; SANFELICE, R. ; TEEL, A. : Hybrid Dynamical Systems. In: *IEEE Control Systems Magazine* 29(2) (2009), S. 28–93
- [63] GOLUB, G. ; VAN LOAN, C. : *Matrix Computations*. Johns Hopkins University Press, 1996
- [64] GOMEZ-SKARMETA, A. ; JIMENEZ, F. : Fuzzy modeling with hybrid systems. In: *Fuzzy Sets and Systems* 104(2) (1999), S. 199–208
- [65] GORRINI, V. ; BERSINI, H. : Recurrent fuzzy systems. In: *Proc. of the IEEE Intern. Conf. on Fuzzy Systems*. Orlando, USA, 1994, S. 193–198

- [66] GRÜNE, L. ; MÜLLER, F. ; JERG, S. ; JUNGE, O. ; POST, M. ; LEHMANN, D. ; LUNZE, J. : Two Complementary Approaches to Event-Based Control. In: *at-Automatisierungstechnik* 58(4) (2010), S. 173–182
- [67] GUILLAUME, S. : Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review. In: *IEEE Trans. on Fuzzy Systems* 9(3) (2001), S. 426–443
- [68] GUILLAUME, S. ; CHARNOMORDIC, B. : Generating an Interpretable Family of Fuzzy Partitions from Data. In: *IEEE Trans. on Fuzzy Systems* 12(3) (2004), S. 324–335
- [69] HAUTH, J. : *Grey-Box Modeling for Nonlinear Systems*, Technische Universität Kaiserslautern, Dissertation, 2008
- [70] HIERSIG, H. (Hrsg.): *Lexikon Produktionstechnik, Verfahrenstechnik*. Springer, 1995
- [71] HIMMELBLAU, D. : Applications of Artificial Neural Networks in Chemical Engineering. In: *Korean Journal of Chemical Engineering* 17(4) (2000), S. 373–392
- [72] HOFFMANN, F. ; NELLES, O. : Genetic programming for model selection of TSK-fuzzy system. In: *Information Science* 136(1) (2001), S. 7–28
- [73] HOMAIFAR, A. ; MCCORMICK, E. : Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms. In: *IEEE Trans. on Fuzzy Systems* 3(2) (1995), S. 129–139
- [74] HONG, X. ; HARRIS, C. : A Neurofuzzy Network Knowledge Extraction and Extended Gram-Schmidt Algorithm for Model Subspace Decomposition. In: *IEEE Trans. on Fuzzy Systems* 11(4) (2003), S. 528–541
- [75] HUANG, K. ; WANG, S.-J. ; IWAKABE, K. ; SHAN, L. ; ZHU, Q. : Temperature control of an ideal heat-integrated distillation column (HIDiC). In: *Chemical Engineering Science* 62(22) (2007), S. 6486–6491
- [76] ILOG INC. (Hrsg.): *CPLEX 6.5 user manual*. ILOG S.A., Gentilly Cedex, France: ILOG Inc., 1999

- [77] ISERMANN, R. : Process fault detection based on modeling and estimation methods - A survey. In: *Automatica* 20(4) (1984), S. 387–404
- [78] ISERMANN, R. : *Identifikation dynamischer Systeme 1*. Springer, 1992
- [79] ISERMANN, R. : *Identifikation dynamischer Systeme 2*. Springer, 1992
- [80] ISERMANN, R. : *Fault Diagnosis Systems - An Introduction from Fault Detection to Fault Tolerance*. Springer, 2006
- [81] ISHIBUCHI, H. ; NAKASHIMA, T. : Genetic-algorithm-based approach to linguistic approximation of nonlinear functions with many input variables. In: *Proc. of the IEEE Intern. Conf. on Fuzzy Systems*. Seoul, Korea, 1999, S. 779–784
- [82] ISHIBUCHI, H. ; NOZAKI, K. ; YAMAMOTO, N. ; TANAKA, H. : Selecting fuzzy if-then rules for classification problems using genetic algorithms. In: *IEEE Trans. on Fuzzy Systems* 3(3) (1995), S. 260–270
- [83] JAKUBEK, S. ; HAMETNER, C. : Identification of Neuro-Fuzzy models using GTLS parameter estimation. In: *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics* 39(5) (2009), S. 1121–1133
- [84] JANG, J. : ANFIS: Adaptive-Network-Based Fuzzy Inference System. In: *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics* 23(3) (1993), S. 665–685
- [85] JARVIS-WLOSZEK, Z. W.: *Lyapunov Based Analysis and Controller Synthesis for Polynomial Systems using Sum-of-Squares Optimization*, University of California, Berkley, Phd Thesis, 2003
- [86] JIN, Y. : Fuzzy Modeling of High-Dimensional Systems: Complexity Reduction and Interpretability Improvement. In: *IEEE Trans. on Fuzzy Systems* 8(2) (2000), S. 212–221
- [87] JOHANSEN, T. : *Operating Regime based Process Modeling and Identification*, University of Trondheim, Dissertation, 1994

- [88] JOHANSEN, T. ; HUNT, K. ; GAWTHROP, P. ; FRITZ, H. : Off-equilibrium linearisation and design of gain scheduled control with application to vehicle speed control. In: *Control Engineering Practice* 6(2) (1998), S. 167–180
- [89] JOHANSSON, M. ; RANTZER, A. : Computation of piecewise quadratic Lyapunov functions for hybrid systems. In: *IEEE Trans. on Automatic Control* 43(4) (1998), S. 555–559
- [90] JUANG, C.-F. : A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design. In: *IEEE Trans. on Systems, Man, And Cybernetics - Part B: Cybernetics* 34(2) (2004), S. 997–1006
- [91] JUANG, C.-F. : Genetic recurrent fuzzy system by coevolutionary computation with divide-and-conquer technique. In: *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics* 35(2) (2005), S. 249–254
- [92] JUANG, C.-F. ; CHANG, P.-H. : Designing Fuzzy-Rule-Based Systems Using Continuous Ant-Colony Optimization. In: *IEEE Trans. on Fuzzy Systems* 18(1) (2010), S. 138–149
- [93] JUANG, C.-F. ; CHUNG, I. ; HSU, C. : Automatic construction of feed-forward/recurrent fuzzy systems by clustering-aided simplex particle swarm optimization. In: *Fuzzy Sets and Systems* 158(18) (2007), S. 1979–1996
- [94] JUANG, C.-F. ; HSIAO, C.-M. ; HSU, C.-H. : Hierarchical Cluster-based Multi-Species Particle Swarm Optimization for Fuzzy System Optimization. In: *IEEE Trans. on Fuzzy Systems* 18(1) (2010), S. 14–26
- [95] JUANG, C.-F. ; LIN, C. : A recurrent self-organizing neural fuzzy inference network. In: *IEEE Trans. on Neural Networks* 10(4) (1999), S. 828–845
- [96] KARNIK, N. ; MENDEL, J. ; LIANG, Q. : Type-2 Fuzzy Logic Systems. In: *IEEE Trans. on Fuzzy Systems* 7(6) (1999), S. 643–658
- [97] KEMPF, R. : *Rekurrente Fuzzy-Systeme*. Fortschrittberichte VDI, Reihe 8, Nr. 1028, VDI Verlag, Düsseldorf, Technische Universität Darmstadt, Dissertation, 2004

- [98] KEMPF, R. ; ADAMY, J. : Equilibria of recurrent fuzzy systems. In: *Fuzzy Sets and Systems* 140(2) (2003), S. 231–257
- [99] KEMPF, R. ; ADAMY, J. : Sequential pattern recognition employing recurrent fuzzy systems. In: *Fuzzy Sets and Systems* 146(3) (2004), S. 451–472
- [100] KENNEDY, J. ; EBERHART, R. ; SHI, Y. : *Swarm Intelligence*. Morgan Kaufmann, 2001
- [101] KHALIL, H. K.: *Nonlinear Systems*. Prentice Hall, 2002
- [102] KISTER, H. : *Distillation operation*. McGraw-Hill, 1990
- [103] KIUPEL, N. ; FRANK, P. : Process supervision with the aid of fuzzy logic. In: *Proc. of the IEEE Intern. Conf. on Systems, Man, and Cybernetics*. Le Touquet, Frankreich, 1993, S. 409–414
- [104] KIUPEL, N. ; KÖPPEN-SELIGER, B. ; KELLINGHAUSEN, H. ; FRANK, P. : Fuzzy residual evaluation concept. In: *Proc. of the IEEE Intern. Conf. on Systems, Man, and Cybernetics*. Vancouver, Kanada, 1995, S. 13–18
- [105] KLATT, K.-U. ; ENGELL, S. : Rührkesselreaktor mit Parallel- und Folgereaktion. In: *Nichtlineare Regelung - Methoden, Werkzeuge, Anwendungen*. VDI-Verlag, 1993, S. 101–108
- [106] KORBICZ, J. ; KOSCIELNY, J. ; KOWALCZUK, Z. ; CHOLEWA, W. : *Fault Diagnosis: Models, Artificial Intelligence, Applications*. Springer, 2004
- [107] KRAUSE, P. ; SLAWINSKI, T. : Das Fuzzy-ROSA-Verfahren: Von der regelorientierten statistischen Analyse zur datenbasierten Generierung von interpretierbaren Takagi-Sugeno-Systemen. In: *at-Automatisierungstechnik* 49(9) (2001), S. 391–399
- [108] KROLL, A. : Identification of functional fuzzy models using multi-dimensional reference fuzzy sets. In: *Fuzzy Sets and Systems* 80(2) (1996), S. 149–158
- [109] KROLL, A. : Grey-box models: Concepts and application. In: MOHAMMADIAN, M. (Hrsg.): *New Frontiers in Computational Intelligence and its Applications* Bd. 52. IOS Press, Amsterdam, 2000, S. 42–51

- [110] KROLL, A. : On the state-space representation of local affine models in Takagi-Sugeno-Kang Fuzzy Systems. In: *at-Automatisierungstechnik* 52 (2004), S. 470–480
- [111] KROLL, A. ; BERND, T. ; TROTT, S. : Fuzzy network model-based fuzzy state controller design. In: *IEEE Trans. on Fuzzy Systems* 8(5) (2000), S. 632–644
- [112] KRUSE, R. ; GEBHARDT, J. ; KLAWONN, F. : *Foundations of Fuzzy Systems*. Wiley, 1994
- [113] LEONHARDT, S. ; AYOUBI, M. : Methods of Fault Diagnosis. In: *Control Engineering Practice* 5(5) (1997), S. 683–692
- [114] LI, C. ; CHENG, K.-H. : Recurrent neuro-fuzzy hybrid-learning approach to accurate system modeling. In: *Fuzzy Sets and Systems* 158(2) (2007), S. 194–212
- [115] LIN, C. ; CHEN, C. : Identification and prediction using recurrent compensatory neuro-fuzzy systems. In: *Fuzzy Sets and Systems* 150(2) (2005), S. 307–330
- [116] LIN, M.-T. ; YU, C.-C. ; LUYBEN, M. : Interpretation of Temperature Control for Ternary Distillation. In: *Industrial & Engineering Chemistry Research* 44(22) (2005), S. 8277–8290
- [117] LIN, Y. ; CUNNINGHAM, G. ; COGGESHALL, S. : Using Fuzzy Partitions to create Fuzzy Systems from Input-Output Data and Set the Initial Weights in a Fuzzy Neural Network. In: *IEEE Trans. on Fuzzy Systems* 5(4) (1997), S. 614–621
- [118] LITH, P. van ; BETLEM, B. ; ROFFEL, B. : Combining prior knowledge with data driven modeling of a batch distillation column including start-up. In: *Computers and Chemical Engineering* 27(7) (2003), S. 1021–1030
- [119] LITTLE, J. ; COX, D. ; O'SHEA, D. : *Using algebraic geometry*. Springer, 2005
- [120] LIU, G. ; YANG, J. ; WHITBORNE, J. : *Multiobjective Optimization and Control*. Research Studies Press, 2002

-
- [121] LÖFBERG, J. : A Toolbox for Modeling and Optimization in MATLAB. In: *Proc. of the CASCD Conf.* Taipeh, Taiwan, 2004, S. 284–289
 - [122] LOPEZ TORIBIO, C. ; PATTON, R. ; DALEY, S. : Takagi-Sugeno Fuzzy Fault-Tolerant Control of an Induction Motor. In: *Neural Computing and Applications* 9(1) (1999), S. 19–28
 - [123] LUENBERGER, D. : *Linear and Nonlinear Programming.* Springer, 2003
 - [124] LUGHOFFER, E. : FLEXFIS: A Robust Incremental Learning Approach for Evolving Takagi-Sugeno Fuzzy Models. In: *IEEE Trans. on Fuzzy Systems* 16(6) (2008), S. 1393–1410
 - [125] LUNZE, J. : Qualitative modelling of dynamical systems: Motivation, methods, and prospective applications. In: *Mathematics and Computers in Simulation* 46 (1998), S. 465–483
 - [126] LUNZE, J. : Diagnosis of Discretely Controlled Continuous Systems. In: *at-Automatisierungstechnik* 54(8) (2006), S. 385–395
 - [127] LUNZE, J. : *Ereignisdiskrete Systeme.* Oldenbourg Verlag, 2006
 - [128] LUO, R. ; SHAO, H. ; ZHANG, Z. : Fuzzy-Neural-Net-Based Inferential Control for a High-Purity Distillation Column. In: *Control Engineering Practice* 3(1) (1995), S. 31–40
 - [129] MACMURRAY, J. ; HIMMELBLAU, D. : Modeling and Control of a Packet Distillation Column Using Artificial Neural Networks. In: *Computers and Chemical Engineering* 19(10) (1995), S. 1077–1088
 - [130] MAMDANI, E. : Application of Fuzzy Logic to Approximate Reasoning using Linguistic Synthesis. In: *IEEE Trans. on Computers* 26(12) (1977), S. 1182–1191
 - [131] MARENBACH, P. : *Rechnergestützte Methoden zur interaktiven Modellierung biotechnologischer Prozesse,* Technische Universität Darmstadt, Dissertation, 1999
 - [132] MASTOROCOSTAS, P. ; THEOCHARIS, J. : A recurrent fuzzy-neural model for dynamic system identification. In: *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics* 32(2) (2002), S. 176–190

- [133] MEJDELL, T. ; SKOGESTAD, S. : Estimation of Distillation Compositions from Multiple Temperature Measurements Using Partial-Least-Squares Regression. In: *Industrial & Engineering Chemistry Research* 30(12) (1991), S. 2543–2555
- [134] MESKIN, N. ; KHORASANI, K. ; RABBATH, C. : A Hybrid Fault Detection and Isolation Strategy for a Network of Unmanned Vehicles in Presence of Large Environmental Disturbances. In: *IEEE Trans. on Control Systems Technology* 18(6) (2010), S. 1422–1429. – im Druck
- [135] MIKUT, R. ; JÄKEL, J. ; GRÖLL, L. : Interpretability issues in data-based learning of fuzzy systems. In: *Fuzzy Sets and Systems* 150(2) (2005), S. 179–197
- [136] MILLER, A. : *Subset Selection in Regression*. Chapman & Hall, 2002
- [137] MIRKIN, B. : *Clustering for Data Mining: A Data Recovery Approach*. Chapman & Hall/CRC, 2005
- [138] MOSER, A. : *Bioprocess technology*. Springer, 1988
- [139] MÜNCHHOF, M. : *Model-Based Fault Detection for a Hydraulic Servo Axis*, Technische Universität Darmstadt, Dissertation, 2006
- [140] NAKOULA, Y. ; GALICHET, S. ; FOULLOY, L. : Simultaneous Learning of Rules and Linguistic Terms. In: *Proc. of the IEEE Intern. Conf. on Fuzzy Systems*. New Orleans, USA, 1996, S. 1743–1749
- [141] NAUCK, D. ; KRUSE, R. : Neuro-fuzzy systems for function approximation. In: *Fuzzy Sets and Systems* 101(2) (1999), S. 261–271
- [142] NELLES, O. : *Nonlinear System Identification*. Springer, 2002
- [143] NÜRNBERGER, A. ; KRUSE, R. : A Neuro-Fuzzy Approach to Optimize Hierarchical Recurrent Fuzzy Systems. In: *Fuzzy Optimization and Decision Making* 1(2) (2002), S. 221–248
- [144] NYBERG, M. : A fault isolation algorithm for the case of multiple faults and multiple fault types. In: *Proc. of the IFAC Symposium on System Identification*. Newcastle, Australien, 2006, S. 303–308
- [145] OLIVEIRA, J. : Semantic Constraints for Membership Function Optimization. In: *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans* 29(1) (1999), S. 128–138

- [146] OLIVEIRA, J. : Towards neuro-linguistic modeling: Constraints for optimization of membership functions. In: *Fuzzy Sets and Systems* 106(3) (1999), S. 357–380
- [147] ORTSEIFEN, A. : *Entwicklung von Verfahren zur Fehlerdiagnose mit rekurrenten Fuzzy-Systemen*, Technische Universität Darmstadt, Diplomarbeit, 2009
- [148] PAPACHRISTODOULOU, A. ; PRAJNA, S. : A Tutorial on Sum of Squares Techniques for System Analysis. In: *Proc. of the American Control Conference*. Portland, USA, 2005, S. 2686–2700
- [149] PARILLO, P. : *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, California Institute of Technology, Pasadena, Phd Thesis, 2000
- [150] PATTON, R. ; CHEN, J. ; LOPEZ TORIBIO, C. : Fuzzy Observers for Non-linear Dynamic Systems Fault Diagnosis. In: *Proc. of the IEEE Intern. Conf. on Decision and Control*. Tampa, USA, 1998, S. 84–89
- [151] PATTON, R. ; CHEN, J. ; SIEW, T. : Fault Diagnosis in Nonlinear Systems via Neural Networks. In: *Proc. of the IEEE Intern. Conf. on Control*. Coventry, Großbritannien, 1994
- [152] PATTON, R. ; UPPAL, F. ; LOPEZ TORIBIO, C. : Soft Computing approaches to fault diagnosis for dynamic systems: A survey. In: *Proc. of the IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes*. Budapest, Ungarn, 2000, S. 298–311
- [153] PEDRYCZ, W. : *Fuzzy Control and Fuzzy Systems*. John Wiley & Sons, New York, 1993
- [154] PFEIFFER, B.-M. ; JÄKEL, J. ; KROLL, A. ; KUHN, C. ; KUNTZE, H.-B. ; LEHMANN, U. ; SLAVINSKI, T. ; TEWS, V. : Erfolgreiche Anwendungen von Fuzzy Logik und Fuzzy Control (Teil 1). In: *at-Automatisierungstechnik* 50 (2002), Nr. 10, S. 461–471
- [155] PFEIFFER, B.-M. ; JÄKEL, J. ; KROLL, A. ; KUHN, C. ; KUNTZE, H.-B. ; LEHMANN, U. ; SLAVINSKI, T. ; TEWS, V. : Erfolgreiche Anwendungen von Fuzzy Logik und Fuzzy Control (Teil 2). In: *at-Automatisierungstechnik* 50 (2002), Nr. 11, S. 511–521

- [156] PFEUFER, T. ; AYOUBI, M. : Application of a hybrid neuro-fuzzy system to the fault diagnosis of an automotive electromechanical actuator. In: *Fuzzy Sets and Systems* 89(3) (1997), S. 351–360
- [157] POMARES, H. ; ROJAS, I. ; GONZALES, J. ; PRIETO, A. : Structure Identification in Complete Rule-Based Fuzzy Systems. In: *IEEE Trans. on Fuzzy Systems* 10(3) (2002), S. 349–359
- [158] PRAJNA, S. ; PAPACHRISTODOULOU, A. : Analysis of switched and hybrid systems - beyond piecewise quadratic methods. In: *Proc. of the American Control Conference*. Denver, USA, 2003, S. 2779–2784
- [159] PRAJNA, S. ; PAPACHRISTODOULOU, A. ; PARRILO, P. : Introducing SOSTOOLS: A general purpose sum of squares programming solver. In: *Proc. of the IEEE Conf. on Decision and Control*. Las Vegas, USA, 2002, S. 741–746
- [160] ROTHFUSS, R. ; RUDOLPH, J. ; ZEITZ, M. : Flatness Based Control of a Nonlinear Chemical Reactor Model. In: *Automatica* 32(10) (1996), S. 1433–1439
- [161] ROUBOS, H. ; SETNES, M. : Compact and Transparent Fuzzy Models and Classifiers Through Iterative Complexity Reduction. In: *IEEE Trans. on Fuzzy Systems* 9(4) (2001), S. 516–524
- [162] RUAN, S. ; ZHOU, Y. ; YU, F. ; PATTIPATI, K. ; WILLETT, P. ; PATTERSON-HINE, A. : Dynamic Multiple-Fault Diagnosis With Imperfect Tests. In: *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans* 39(6) (2009), S. 1224–1236
- [163] RUSSO, M. : FuGeNeSys - A Fuzzy Genetic Neural System for Fuzzy Modeling. In: *IEEE Trans. on Fuzzy Systems* 6(3) (1998), S. 373–388
- [164] SAHINIDIS, N. : BARON - Branch and reduce optimization navigator / University of Illinois at Urbana-Champaign, Department of Chemical Engineering, Urbana, USA. 2000. – Forschungsbericht
- [165] SAMPATH, M. ; SENGUPTA, R. ; LAFORTUNE, S. ; SINNAMOHIDEEN, K. ; TENEKETZIS, D. : Diagnosability of Discrete-Event Systems. In: *IEEE Trans. on Automatic Control* 40(9) (1995), S. 1555–1575
- [166] SAMPATH, M. ; SENGUPTA, R. ; LAFORTUNE, S. ; SINNAMOHIDEEN, K. ; TENEKETZIS, D. : Failure Diagnosis Using Discrete-Event Models. In: *IEEE Trans. on Control Systems Technology* 4(2) (1996), S. 105–124

-
- [167] SAUTER, D. ; MARY, N. ; SIROU, F. ; THIELTGEN, A. : Fault diagnosis in systems using fuzzy logic. In: *Proc. of the IEEE Conf. on Control Applications*. Glasgow, Großbritannien, 1994, S. 883–888
- [168] SCHNEIDER, H. ; FRANK, P. : Observed-Based Supervision and Fault Detection in Robots Using Nonlinear and Fuzzy Logic Residual Evaluation. In: *IEEE Trans. on Control Systems Technology* 4(3) (1996), S. 274–282
- [169] SCHREIBER, S. : *Subset Selection zur nichtlinearen Identifikation mit rekurrenten Fuzzy Systemen*, Technische Universität Darmstadt, Studienarbeit, 2010
- [170] SCHULLERUS, G. ; SUPAVATANAKUL, P. ; KREBS, V. ; LUNZE, J. : Diagnose zeitbewerteter ereignisdiskreter Systeme. In: *at-Automatisierungstechnik* 52(4) (2004), S. 164–173
- [171] SCHWEFEL, H.-P. : *Evolution and Optimum Seeking*. John Wiley & Sons, 1995
- [172] SCHWUNG, A. ; ADAMY, J. : Nonlinear System Modeling via Hybrid System Representation of Recurrent Fuzzy Systems. In: *Proc. of the IEEE Intern. Conf. on Fuzzy Systems*. Barcelona, Spanien, 2010, S. 2381–2387
- [173] SCHWUNG, A. ; GUSSNER, T. ; ADAMY, J. : Stability Analysis of Continuous-time Recurrent Fuzzy Systems. In: *Proc. of the IEEE Intern. Conf. on Fuzzy Systems*. Jeju, Korea, 2009, S. 313–318
- [174] SCHWUNG, A. ; GUSSNER, T. ; ADAMY, J. : Stability Analysis of Recurrent Fuzzy Systems: A hybrid system and SOS approach. In: *IEEE Trans. on Fuzzy Systems* 19(3) (2011). – im Druck
- [175] SCHWUNG, A. ; ORTSEIFEN, A. ; ADAMY, J. : Anwendung zeitdiskreter rekurrenter Fuzzy Systeme zur Fehlerdiagnose. In: *at-Automatisierungstechnik* 58(6) (2010), S. 322–331
- [176] SETNES, M. ; BABUŠKA, R. : Rule Base Reduction: Some Comments on the Use of Orthogonal Transforms. In: *IEEE Trans. on Systems, Man, Cybernetics - Part C: Applications and Reviews* 31(2) (2001), S. 199–206

- [177] SETNES, M. ; BABUŠKA, R. ; VERBRUGGEN, H. : Rule-Based Modeling: Precision and Transparency. In: *IEEE Trans. on Systems, Man and Cybernetics - Part C: Applications and Reviews* 28(1) (1998), S. 165–169
- [178] SHAKERI, M. ; RAGHAVAN, V. ; PATTIPATI, K. ; PATTERSON-HINE, A. : Sequential Testing Algorithms for Multiple Fault Diagnosis. In: *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans* 30(1) (2000), S. 1–14
- [179] SHAW, A. ; DOYLE, F. : Multivariable Nonlinear Control Applications for a High Purity Distillation Column Using a Recurrent Dynamic Neuron Model. In: *Journal of Process Control* 7(4) (1997), S. 255–268
- [180] SKOGESTAD, S. : Dynamics and Control of Distillation Columns - A Critical Survey. In: *Proc. of the IFAC Symp. on Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes*. Maryland, USA, 1992, S. 1–25
- [181] SLAWINSKI, T. ; KRONE, A. ; KRAUSE, P. ; KIENDL, H. : The fuzzy-ROSA method: A statistically motivated fuzzy approach for data-based generation of small interpretable rule bases in high-dimensional search spaces. In: *Data mining and computational intelligence* (2001), S. 141–166
- [182] SUBRAMANIAN, S. ; MOONEY, R. : Qualitative Multiple-fault Diagnosis of Continuous Dynamic Systems Using Behavioral Modes. In: *Proc. of the National Conf. on Artificial Intelligence*. Menlo Park, USA, 1996
- [183] SUGENO, M. : On Stability of Fuzzy Systems expressed by Fuzzy Rules with Singleton Consequents. In: *IEEE Trans. on Fuzzy Systems* 7(2) (1999), S. 201–224
- [184] SUGENO, M. ; YASUKAWA, T. : A Fuzzy-Logic-Based Approach to Qualitative Modeling. In: *IEEE Trans. on Fuzzy Systems* 1(1) (1993), S. 7–31
- [185] SUPAVATANAKUL, P. ; LUNZE, J. ; PUIG, V. ; QUEVEDO, J. : Diagnosis of timed automata: Theory and application to the DAMADICS actuator benchmark problem. In: *Control Engineering Practice* 14(6) (2006), S. 609–619

- [186] SURMANN, H. ; MANIADAKIS, M. : Learning feed-forward and recurrent fuzzy systems: A genetic approach. In: *Journal of Systems Architecture* 47(7) (2001), S. 649–662
- [187] TAKAGI, T. ; SUGENO, M. : Fuzzy Identification of Systems and Its Application to Modeling and Control. In: *IEEE Trans. on Systems, Man, and Cybernetics* 15(1) (1985), Nr. 1, S. 116–132
- [188] TANAKA, K. ; SANO, M. : A Robust Stabilization Problem of Fuzzy Control Systems and Its Application to Backing up Control of a Truck-Trailer. In: *IEEE Trans. on Fuzzy Systems* 2(2) (1994), S. 119–134
- [189] TANAKA, K. ; WANG, H. : *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. John Wiley & Sons, 2001
- [190] TANAKA, K. ; YOSHIDA, H. ; OHTAKE, H. ; WANG, H. : A Sum of Squares Approach to Stability Analysis of Polynomial Fuzzy Systems. In: *Proc. of the American Control Conference*. New York, USA, 2007, S. 4071–4076
- [191] TAWARMALANI, M. ; SAHINIDIS, N. : *Convex and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Springer, 2003
- [192] THEOCHARIS, J. : A high-order recurrent neuro-fuzzy system with internal dynamics: Application to the adaptive noise cancellation. In: *Fuzzy Sets and Systems* 157(4) (2006), S. 471–500
- [193] TOH, K. ; TODD, M. ; TUTUNCU, R. : SDPT3 - a Matlab software package for semidefinite programming. In: *Optimization Methods and Software* 11 (1999), S. 545–581
- [194] TOPCU, U. ; PACKARD, A. ; SAILER, P. ; BALAS, G. : Help on SOS [Ask the Experts]. In: *IEEE Control Systems Magazine* 30(4) (2010), S. 18–23
- [195] TU, F. ; PATTIPATI, K. ; S., D. ; MALEPATI, V. : Computationally Efficient Algorithms for Multiple Fault Diagnosis With Unreliable Tests. In: *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans* 33(1) (2003), S. 73–85

-
- [196] VAVATZANIDOU, A. : *Strukturoptimierung rekurrenter Fuzzy Systeme mit genetischer Programmierung*, Technische Universität Darmstadt, Bachelorarbeit, 2009
- [197] VEDAM, H. ; VENKATASUBRAMANIAN, V. : Signed Digraph Based Multiple Fault Diagnosis. In: *Computers and Chemical Engineering* 21(6) (1997), S. 655–660
- [198] VENKATASUBRAMANIAN, V. ; RENGASWAMY, R. ; KAVURI, S. : A review of process fault detection and diagnosis. Part 2: Qualitative models and search strategies. In: *Computers and Chemical Engineering* 27(3) (2003), S. 313–326
- [199] VENKATASUBRAMANIAN, V. ; RENGASWAMY, R. ; KAVURI, S. ; YIN, K. : A review of process fault detection and diagnosis. Part 3: Process history based methods. In: *Computers and Chemical Engineering* 27(3) (2003), S. 327–346
- [200] VENKATASUBRAMANIAN, V. ; RENGASWAMY, R. ; YIN, K. ; KAVURI, S. : A review of process fault detection and diagnosis. Part 1: Quantitative model-based methods. In: *Computers and Chemical Engineering* 27(3) (2003), S. 293–311
- [201] VOIGT, C. ; ADAMY, J. : *Formelsammlung der Matrizenrechnung*. Oldenbourg Verlag, 2007
- [202] WAKI, H. ; KIM, S. ; KOJIMA, M. ; MURAMATSU, M. : Sums of Squares and Semidefinite Programming Relaxations for Polynomial Optimization Problems with Structured Sparsity / Tokyo Institute of Technology, Japan. 2005. – Forschungsbericht
- [203] WANG, L.-X. : The WM Method Completed: A Flexible Fuzzy System Approach to Data Mining. In: *IEEE Trans. on Fuzzy Systems* 11(6) (2003), S. 768–783
- [204] WANG, L.-X. ; MENDEL, J. : Generating Fuzzy Rules by Learning from Examples. In: *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics* 22(6) (1992), S. 1414–1427
- [205] WANG, L.-X. ; MENDEL, J. : Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least-Squares Learning. In: *IEEE Trans. on Neural Networks* 3(5) (1992), S. 807–814

- [206] WELD, D. ; DEKLEER, J. : *Readings in Qualitative Reasoning*. Morgan Kaufmann, 1990
- [207] WERBOS, P. : Backpropagation Through Time: What It Does and How To Do It. In: *Proc. of the IEEE* 78(10) (1990), S. 1550–1560
- [208] WILLIMOWSKI, M. ; FÜSSEL, D. ; ISERMANN, R. : Diagnose von Verbrennungsaussetzern in Ottomotoren durch Messung des Abgasdrucks. In: *mtz - Motortechnische Zeitschrift* 60(10) (1999), S. 654–663
- [209] WOLSEY, L. : *Integer Programming*. John Wiley & Sons, 1998
- [210] YAM, Y. ; BARANYI, P. ; YANG, C.-T. : Reduction of Fuzzy Rule Base Via Singular Value Decomposition. In: *IEEE Trans. on Fuzzy Systems* 7(2) (1999), S. 120–132
- [211] YEN, J. ; WANG, L. : Application of Statistical Information Criteria for Optimal Fuzzy Model Construction. In: *IEEE Trans. on Fuzzy Systems* 6(3) (1998), S. 362–372
- [212] ZADEH, L. : Optimality and non-scalar-valued performance criteria. In: *IEEE Trans. on Automatic Control* 8(1) (1963), S. 59–60
- [213] ZADEH, L. : The concept of a linguistic variable and its application to approximate reasoning I. In: *Information Science* 8(3) (1975), S. 199–251
- [214] ZENG, X. ; SINGH, M. : Approximation Theory of Fuzzy Systems - MIMO Case. In: *IEEE Trans. on Fuzzy Systems* 3(2) (1995), S. 219–235
- [215] ZENG, X. ; SINGH, M. : Decomposition Property of Fuzzy Systems and its Application. In: *IEEE Trans. on Fuzzy Systems* 4(2) (1996), S. 149–165
- [216] ZHANG, J. ; MORRIS, A. : Recurrent Neuro-Fuzzy Networks for Non-linear Process Modeling. In: *IEEE Trans. on Neural Networks* 10(2) (1999), S. 313–326
- [217] ZHOU, S. ; GAN, J. : Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling. In: *Fuzzy Sets and Systems* 159(23) (2008), S. 3091–3131

-
- [218] ZHUANG, Z. ; FRANK, P. : Qualitative Observers and its application to fault detection and isolation systems. In: *Proc. of the Institution of Mechanical Engineers, Part 1* 211(4) (1997), S. 253–262